

**PENGEMBANGAN SISTEM INFORMASI PELAYANAN SURAT
KETERANGAN STUDI KASUS: PEMERINTAH DESA LEGUNDI
KECAMATAN KARANGJATI KABUPATEN NGAWI**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Tio Renndy Winarna
NIM: 145150400111102



**PROGRAM STUDI SISTEM INFORMASI
JURUSAN SISTEM INFORMASI
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018**

PENGESAHAN

PENGEMBANGAN SISTEM INFORMASI PELAYANAN SURAT KETERANGAN STUDI
KASUS: PEMERINTAH DESA LEGUNDI KECAMATAN KARANGJATI KABUPATEN
NGAWI

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :

Tio Renndy Winarna

NIM: 145150400111102

Skripsi ini telah diuji dan dinyatakan lulus pada
25 Juli 2018

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II



Ismiarta Aknuranda, S.T, M.Sc, Ph.D

NIK: 201006 740719 1 001



Moch. Chandra Saputra, S.Kom, M.T, M.Eng.

NIK: 201609 860106 1 001



Mengetahui

Ketua Jurusan Sistem Informasi

Dr. Eng., Herman Tolle, S.T, M.T.

NIP: 19740823 200012 1 001

IDENTITAS TIM PENGUJI

Dosen Penguji I

Dosen Penguji II

Welly Purnomo, S.T., M.Kom.

NIK: 201708 810117 1 001

Alfi Nur Rusydi, S.Si., M.Sc.

NIK: 201607 880412 1 001





TIO RENNDY WINARNA

Tempat, Tanggal Lahir
Ngawi, 07 Juni 1996

Jenis Kelamin

Laki-laki

Agama

Islam

Kewarganegaraan

Indonesia

Status

Belum Kawin

Alamat

Desa Legundi RT 08 RW 01

Kecamatan Karangjati

Kabupaten Ngawi



+62895339168911



tiorenndy@gmail.com



ti_renn

CURRICULUM VITAE

Pendidikan

- 2014 - Sekarang : Fakultas Ilmu Komputer
Universitas Brawijaya
- 2011 - 2014 : SMAN 1 Mejayan
- 2008 - 2011 : SMPN 1 Karangjati
- 2002 - 2008 : SDN Legundi 2

Pengalaman Organisasi

- 2014 : Wakil Menteri Kominfo FORMADIMA
(Forum Mahasiswa Madiun Studi Malang)
- 2015 : Staff Departemen Advokesma EMSI
(Eksekutif Mahasiswa Sistem Informasi)
- 2015 : Wakil Ketua Umum FORMADIMA
(Forum Mahasiswa Madiun Studi Malang)
- 2016 : Staff Departemen Kewirausahaan EMSI
(Eksekutif Mahasiswa Sistem Informasi)

Kemampuan

- Microsoft Office Word, Exel, Power Point, Access & Outlook
- HTML, PHP, CSS, JAVA, SQL
- Bahasa Indonesia (aktif), Inggris (pasif)

Hobbi



Sepakbola



Musik

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 25 Juli 2018



Tio Renndy Winarna

NIM: 145150400111102

KATA PENGANTAR

Segala puji bagi Allah SWT atas limpahan rahmat yang diberikan sehingga penulis dapat menyelesaikan skripsi berjudul “Pengembangan Sistem Informasi Pelayanan Surat Keterangan Studi Kasus: Pemerintah Desa Legundi Kecamatan Karangjati Kabupaten Ngawi”.

Penulis menyadari bahwa skripsi ini tidak akan berhasil tanpa bantuan dari beberapa pihak. Oleh karena itu, penulis ingin menyampaikan rasa hormat dan terima kasih kepada:

1. Bapak Ismiarta Aknuranda, S.T, M.Sc, Ph.D, selaku pembimbing satu yang telah mengajarkan ketegasan dan kedisiplinan dalam menyelesaikan kewajiban. Memberikan konsep pemikiran yang luas dalam mencari solusi dari sebuah permasalahan.
2. Bapak Mochamad Chandra Saputra, S.Kom, M.Eng, selaku pembimbing dua yang mengajarkan ketegasan dan kedisiplinan dalam menyelesaikan kewajiban. Memberikan contoh sebagai pribadi yang sabar, solutif, dan profesional dalam memberikan bimbingan kepada mahasiswa.
3. Wayan Firdaus Mahmudy, S.Si, M.T, Ph.D selaku Dekan Fakultas Ilmu Komputer Universitas Brawijaya.
4. Dr. Eng. Herman Tolle, S.T, M.T selaku Ketua Jurusan Sistem Informasi Fakultas Ilmu Komputer Universitas Brawijaya.
5. Suprpto, S.T, M.T selaku Ketua Program Studi Sistem Informasi Jurusan Sistem Informasi Fakultas Ilmu Komputer Universitas Brawijaya.
6. Ibu, Ayah dan kakak-kakak saya yang selalu memberikan dukungan doa, moril, materiil, mengajarkan ketegasan, kedisiplinan dalam menyelesaikan tugas, dan senantiasa memberikan contoh yang baik untuk menjadi pribadi yang sabar dan pekerja keras.
7. Bapak Sumarsono selaku Sekretaris Desa Legundi yang memberikan kesempatan dan kemudahan untuk dapat melakukan penelitian di lokasi studi kasus. Yang memberikan contoh untuk senantiasa bekerja profesional dan mengamalkan ilmu yang dimiliki untuk dapat membantu masyarakat.
8. Penduduk Desa Legundi yang telah memberikan waktu dan kesempatan guna membantu penelitian ini.
9. Ayu Meilinda Puspitasari, *The Special One* yang telah membantu dan selalu memberikan doa, semangat dan dukungan dalam menyelesaikan penelitian ini.
10. Teman-teman Jurusan Sistem Informasi, Fakultas Ilmu Komputer, Preman Kalem, Goyang Piranha, Kembang Kertas 28A, CP27, Formadima, AdIns Malang, Autis serta teman-teman yang lain yang tidak dapat saya sebutkan

sebutkan namanya satu per satu, yang telah memberikan saya dukungan moril, menjadi teman dan sahabat yang baik dan menyenangkan selama masa perkuliahan.

Semoga Allah SWT melimpahkan rahmat berlipat ganda kepada seluruh pihak yang membantu dan berkontribusi pada proses penelitian ini. Tidak lupa saya sampaikan bahwa penelitian yang telah dilakukan memiliki banyak kekurangan, diskusi mengenai penelitian lanjutan, saran, dan kritik yang membangun merupakan beberapa hal yang saya harapkan dapat disampaikan kepada saya.

Malang, 25 Juli 2018

Penulis

tiorenndy@gmail.com



ABSTRAK

Tio Renndy Winarna, Pengembangan Sistem Informasi Pelayanan Surat Keterangan Studi Kasus: Pemerintah Desa Legundi Kecamatan Karangjati Kabupaten Ngawi

Dosen Pembimbing: Ismiarta Aknuranda, S.T, M.Sc, Ph.D dan Moch. Chandra Saputra, S.Kom, M.T, M.Eng.

Desa Legundi adalah desa yang berada pada wilayah Kecamatan Karangjati Kabupaten Ngawi Jawa Timur. Dalam kasus Pemerintah Desa Legundi, pelayanan publik seperti pengurusan dokumen Surat Keterangan masih menemui banyak permasalahan dan untuk mengajukan permohonan surat keterangan penduduk masih harus datang ke kantor desa. Untuk menyelesaikan masalah tersebut diperlukan alat bantu sistem informasi yang mengatasi permohonan surat keterangan dan dokumentasi surat keterangan. Penelitian ini dilakukan untuk mengetahui pengembangan Sistem Informasi Pelayanan Surat Keterangan dapat dijalankan. Untuk menjawab masalah tersebut, maka dilakukan pemodelan proses bisnis, analisis persyaratan, perancangan dan implementasi sistem informasi pelayanan surat keterangan dengan menerapkan pendekatan berorientasi objek. Setelah itu, hasil implementasi sistem diuji untuk mengetahui sistem dapat berjalan sesuai dengan persyaratan yang telah teridentifikasi. Kegiatan pemodelan proses bisnis menghasilkan model proses bisnis As-Is dan proses bisnis To-Be, analisis persyaratan menghasilkan identifikasi persyaratan sistem, serta visualisasi kemampuan sistem dalam diagram use case. Hasil perancangan sistem merupakan dokumentasi model interaksi objek ke dalam diagram sequence, model objek dalam diagram kelas, rancangan basis data, rancangan antarmuka dan pseudocode. Hasil implementasi merupakan Sistem Informasi Pelayanan Surat Keterangan berbasis web yang diuji dan menunjukkan hasil pengujian bahwa sistem dapat berjalan sesuai dengan persyaratan yang teridentifikasi sebelumnya.

Kata kunci: Sistem Informasi, Pelayanan publik, Pelayanan surat keterangan, Desa Legundi

ABSTRACT

Tio Renndy Winarna, Pengembangan Sistem Informasi Pelayanan Surat Keterangan Studi Kasus: Pemerintah Desa Legundi Kecamatan Karangjati Kabupaten Ngawi

Supervisors: Ismiarta Aknuranda, S.T, M.Sc, Ph.D and Moch. Chandra Saputra, S.Kom, M.T, M.Eng.

Legundi village is located in Kecamatan Karangjati Kabupaten Ngawi East Java. In Legundi government case, public service such as Certificate document administration is still has a lot of troubles and to apply for a Certificate document the applicant must come to the village office. To finish the problem is needed an instrument information system that resolve the certificate and others documents. This research done to know the development of Sistem Informasi Pelayanan Surat Keterangan can be run. To solve the problem it has to do the business process modeling, requirements analysis, design and implementation of information service system certificate by applying an object-oriented approach. After that the implementation of the system will be tested to find does the system can run as the requirements that have been identified. Business process modeling activities produce the As-Is business process model and To-Be business process model, the requirements analysis yields identification of system requirements, as well as visualization of system capabilities in use case diagrams. The result of system design constitutes the documentation of object interaction model into sequence diagram, object model in class diagram, database design, interface design and pseudocode. The result of the implementation constitutes a web-based of Sistem Informasi Pelayanan Surat Keterangan tested and shows the results that the system can run in accordance with the requirement that has been identified previously.

Keywords: *Information system, Public services, Certificate document service, Legundi village*

DAFTAR ISI

PENGESAHAN	ii
IDENTITAS TIM PENGUJI.....	iii
DAFTAR RIWAYAT HIDUP	iv
PERNYATAAN ORISINALITAS.....	v
KATA PENGANTAR	vi
ABSTRAK	viii
ABSTRACT	ix
DAFTAR ISI	x
DAFTAR TABEL	xv
DAFTAR GAMBAR	xviii
BAB 1 PENDAHULUAN.....	20
1.1 Latar belakang.....	20
1.2 Rumusan masalah	22
1.3 Batasan Masalah	22
1.4 Tujuan Penelitian	22
1.5 Manfaat	23
1.6 Sistematika Penulisan.....	23
BAB 2 LANDASAN KEPUSTAKAAN	25
2.1 Kajian Pustaka	25
2.2 Pelayanan Publik	26
2.3 Pemerintah Desa Legundi.....	26
2.4 Pengembangan Perangkat Lunak dengan Model <i>Waterfall</i>	27
2.5 Sistem Informasi	28
2.6 Proses Bisnis.....	28
2.6.1 Analisis dan Pemodelan Bisnis As-Is	29
2.6.2 Analisis dan Pemodelan Proses Bisnis To-Be	29
2.7 <i>User Story</i>	30
2.8 <i>Business Process Modeling Notation (BPMN)</i>	30

2.8.1	<i>Flow Object</i>	30
2.8.2	<i>Data</i>	32
2.8.3	<i>Connecting Object</i>	33
2.8.4	<i>Swimlanes</i>	34
2.8.5	<i>Artifacts</i>	34
2.9	Pemodelan <i>Use Case</i>	34
2.9.1	Identifikasi Tipe Pemangku Kepentingan	35
2.9.2	Analisis Masalah	35
2.9.3	Identifikasi Kebutuhan Pemangku Kepentingan dan Pengguna	36
2.9.4	Identifikasi Fitur	36
2.9.5	Identifikasi Persyaratan Fungsional dan Persyaratan Nonfungsional	37
2.9.6	Pemodelan <i>Use Case</i>	38
2.10	<i>Unified Modeling Language (UML)</i>	39
2.10.1	<i>Use Case Diagram</i>	39
2.10.2	<i>Sequence Diagram</i>	40
2.10.3	<i>Class Diagram</i>	43
2.11	<i>Physical Data Model</i>	44
2.12	<i>Model-View-Controller</i>	45
2.13.1	2.13 Pengujian Perangkat Lunak	45
2.13.2	Pengujian Validasi	45
2.13.3	Pengujian Responsif	46
	Pengujian Kompatibilitas	46
BAB 3	METODOLOGI	47
3.1	Studi Pustaka	47
3.2	Pengumpulan Data	48
3.3	Pemodelan Proses Bisnis	48
3.4	Analisis Persyaratan	49
3.5	Perancangan	49
3.6	Implementasi	49

3.7 Pengujian	49
3.8 Penarikan Kesimpulan dan Saran.....	50
BAB 4 Pemodelan proses bisnis dan ANALISIS persyaratan sistem	51
4.1 Pemodelan Proses Bisnis	51
4.1.1 Proses Bisnis <i>As-Is</i>	51
4.1.2 Proses Bisnis <i>To-Be</i>	52
4.2 Analisis Persyaratan	55
4.2.1 Identifikasi Tipe Pemangku Kepentingan.....	55
4.2.2 Analisis Masalah	56
4.2.3 Identifikasi Kebutuhan Pemangku Kepentingan dan Pengguna .	57
4.2.4 Identifikasi Pengguna.....	59
4.2.5 Identifikasi Fitur.....	60
4.2.6 Persyaratan Fungsional.....	62
4.2.7 Persyaratan Non Fungsional	65
4.3 Pemodelan <i>Use Case</i>	65
4.3.1 <i>Use Case</i> Diagram	66
4.3.2 Deskripsi Aktor.....	68
4.3.3 Spesifikasi <i>Use Case</i>	68
BAB 5 PERANCANGAN.....	84
5.1 Perancangan Interaksi Objek.....	84
5.1.1 Mengubah Status Surat Keterangan.....	84
5.1.2 Memeriksa Status Surat Keterangan	84
5.1.3 Menyimpan Surat Keterangan	85
5.1.4 Menghapus Surat Keterangan.....	86
5.2 Pemodelan Class	86
5.3 Perancangan Basis Data	90
5.3.1 Tabel Header	91
5.3.2 Tabel Detail Kutipan Kelahiran	92
5.3.3 Tabel Detail Belum Menikah	92
5.3.4 Tabel Detail Kelahiran	93

5.3.5 Tabel Detail Ktp Proses	93
5.3.6 Tabel Detail Kematian	94
5.3.7 Tabel Jabatan	95
5.3.8 Tabel Jenis Kelamin	95
5.3.9 Tabel Jenis Surat	95
5.3.10 Tabel Status Kawin	95
5.3.11 Tabel Pengguna	96
5.3.12 Tabel Bidang	96
5.3.13 Tabel Status Surat	96
5.4 Perancangan Algoritme	97
5.4.1 Menyimpan Surat Keterangan	97
5.4.2 Menghapus Surat Keterangan	99
5.4.3 Mengubah Status Surat Keterangan	100
5.4.4 Memeriksa Status Surat Keterangan	100
5.5 Perancangan Antarmuka	101
5.5.1 <i>Login</i>	101
5.5.2 Admin	102
5.5.3 Beranda	102
5.5.4 Kelola Bidang	103
5.5.5 Pengaturan	103
BAB 6 Implementasi	105
6.1 Spesifikasi Lingkungan Implementasi	105
6.1.1 Perangkat Keras	105
6.1.2 Perangkat Lunak	105
6.1.3 Minimal Lingkungan <i>Deployment</i>	106
6.2 Implementasi Algoritme	106
6.2.1 Menyimpan Surat Keterangan	106
6.2.2 Menghapus Surat Keterangan	109
6.2.3 Mengubah Status Surat Keterangan	109
6.2.4 Memeriksa Status Surat Keterangan	110

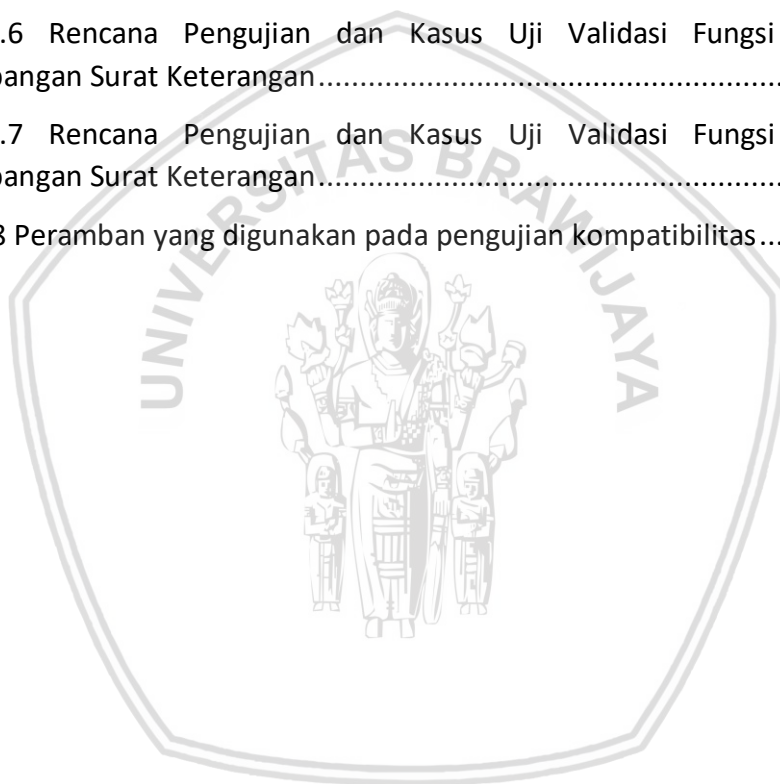
6.3 Implementasi Antarmuka	111
6.3.1 <i>Login</i>	111
6.3.2 Admin	112
6.3.3 Beranda	112
6.3.4 Kelola Bidang	113
6.3.5 Pengaturan	113
BAB 7 Pengujian.....	115
7.1 Pengujian Validasi	115
7.1.1 Pengujian Validasi Memohon Surat Keterangan.....	115
7.1.2 Pengujian Validasi Memeriksa Perkembangan Permohonan ...	118
7.2 Pengujian Responsif	120
7.2.1 Resolusi Layar 1366px × 768px 14 inci.....	120
7.2.2 Resolusi Layar 1080px × 1920px 5.5 inci.....	121
7.3 Pengujian Kompatibilitas	122
BAB 8 Penutup.....	124
8.1 Kesimpulan	124
8.2 Saran	125
DAFTAR PUSTAKA	126
LAMPIRAN A MODEL PROSES BISNIS	128
A.1 Proses Bisnis <i>As-Is</i> Permohonan Surat Keterangan	128
A.2 Proses Bisnis <i>To-Be</i> Permohonan Surat Keterangan	129
LAMPIRAN B HASIL WAWANCARA	130
B.1 Hasil Wawancara Dengan Sekretaris Desa Legundi.....	130
B.2 Hasil Wawancara Dengan Penduduk Desa Legundi	132

DAFTAR TABEL

Tabel 2.1 Penelitian Terdahulu.....	25
Tabel 2.2 Notasi <i>Event</i>	31
Tabel 2.3 Notasi <i>Activity</i>	31
Tabel 2.4 Notasi <i>Gateway</i>	32
Tabel 2.5 Notasi Data	33
Tabel 2.6 Notasi <i>Connecting Objects</i>	33
Tabel 2.7 Notasi <i>Swimlanes</i>	34
Tabel 2.8 Notasi <i>Artifacts</i>	34
Tabel 2.9 Kerangka Dokumentasi Pernyataan Masalah	36
Tabel 2.10 Contoh Tabel Fitur	37
Tabel 2.11 Format Dokumentasi Spesifikasi <i>Use Case</i>	38
Tabel 2.12 Notasi pada <i>Use Case Diagram</i>	39
Tabel 2.13 Notasi pada <i>Sequence Diagram</i>	41
Tabel 2.14 Notasi pada <i>Class Diagram</i>	43
Tabel 4.1 Perubahan Aktivitas Proses Bisnis Permohonan Surat Keterangan	53
Tabel 4.2 Perubahan Aktivitas Proses Bisnis Periksa Hasil Permohonan Surat Keterangan	54
Tabel 4.3 Tipe Pemangku Kepentingan.....	56
Tabel 4.4 Pernyataan Masalah 1	57
Tabel 4.5 Pernyataan Masalah 2	57
Tabel 4.6 Identifikasi Kebutuhan Pengguna.....	58
Tabel 4.7 Identifikasi Pengguna.....	60
Tabel 4.8 Hasil Identifikasi Fitur	61
Tabel 4.9 Hubungan Fitur dengan Kebutuhan Pengguna	62
Tabel 4.10 Persyaratan Fungsional.....	63
Tabel 4.11 Persyaratan Non Fungsional	65
Tabel 4.12 Hubungan Aktivitas Proses Bisnis <i>To-Be</i> dengan <i>Use Case</i>	67
Tabel 4.13 Hubungan <i>Use Case</i> dengan Pemangku Kepentingan	67

Tabel 4.14 Hubungan <i>Use Case</i> dengan Fitur	67
Tabel 4.15 Deskripsi Aktor	68
Tabel 4.16 Spesifikasi <i>Use Case</i> Login	68
Tabel 4.17 Spesifikasi <i>Use Case</i> Mengatur Data Pengguna	70
Tabel 4.18 Spesifikasi <i>Use Case</i> Memohon surat keterangan	72
Tabel 4.19 Spesifikasi <i>Use Case</i> Memeriksa Perkembangan Permohonan	81
Tabel 4.20 Spesifikasi <i>Use Case</i> Membuat Arsip Data Registrasi	82
Tabel 5.1 Struktur Tabel Header	91
Tabel 5.2 Tabel Detail Kutipan Kelahiran	92
Tabel 5.3 Tabel Detail Belum Menikah	93
Tabel 5.4 Tabel Detail Kelahiran	93
Tabel 5.5 Tabel Detail Ktp Proses	93
Tabel 5.6 Tabel Detail Kematian	94
Tabel 5.7 Tabel Jabatan	95
Tabel 5.8 Tabel Jenis Kelamin	95
Tabel 5.9 Tabel Jenis Surat	95
Tabel 5.10 Tabel Status Kawin	96
Tabel 5.11 Tabel Pengguna	96
Tabel 5.12 Tabel Bidang	96
Tabel 5.13 Algoritme Menyimpan Surat Keterangan	97
Tabel 5.14 Algoritme Menghapus Surat Keterangan	99
Tabel 5.15 Algoritme Mengubah Status Surat Keterangan	100
Tabel 5.16 Algoritme Memeriksa Status Surat Keterangan	100
Tabel 6.1 Spesifikasi lingkungan perangkat keras	105
Tabel 6.2 Spesifikasi lingkungan perangkat lunak	105
Tabel 6.3 Spesifikasi Minimal Lingkungan <i>Deployment</i>	106
Tabel 6.4 Implementasi Menyimpan Surat Keterangan	107
Tabel 6.5 Implementasi Menghapus Surat Keterangan	109
Tabel 6.6 Implementasi Mengubah Status Surat Keterangan	110
Tabel 6.7 Implementasi Memeriksa Status Surat Keterangan	110

Tabel 7.1 Matriks Skenario Pengujian Validasi Memohon Surat Keterangan	115
Tabel 7.2 Rencana Pengujian dan Kasus Uji Validasi Fungsi Memohon Surat Keterangan	116
Tabel 7.3 Rencana Pengujian dan Kasus Uji Validasi Fungsi Mengubah Status Surat Keterangan	117
Tabel 7.4 Rencana Pengujian dan Kasus Uji Validasi Fungsi Menghapus Surat Keterangan	118
Tabel 7.5 Matriks Skenario Pengujian Validasi Memeriksa Perkembangan Surat Keterangan	118
Tabel 7.6 Rencana Pengujian dan Kasus Uji Validasi Fungsi Memeriksa Perkembangan Surat Keterangan.....	119
Tabel 7.7 Rencana Pengujian dan Kasus Uji Validasi Fungsi Memeriksa Perkembangan Surat Keterangan.....	119
Tabel 7.8 Peramban yang digunakan pada pengujian kompatibilitas.....	122



DAFTAR GAMBAR

Gambar 2.1 Model <i>Waterfall</i>	27
Gambar 2.2 Sistem Informasi	28
Gambar 2.3 Contoh <i>Use Case Diagram</i>	39
Gambar 2.4 Contoh <i>Sequence Diagram</i>	41
Gambar 2.5 Contoh <i>Physical Data Model</i>	44
Gambar 3.1 Diagram Alur Metode Penelitian	47
Gambar 4.1 Proses Bisnis <i>As-Is</i> Periksa Perkembangan Permohonan Surat Keterangan	52
Gambar 4.2 Proses Bisnis <i>To-Be</i> Periksa Hasil Permohonan Surat Keterangan	55
Gambar 4.3 Use Case Diagram Sistem Informasi Pelayanan Surat Keterangan Pemerintah Desa Legundi	66
Gambar 5.1 Sequence Diagram Mengubah Status Surat Keterangan	84
Gambar 5.2 Sequence Diagram Memeriksa Status Surat Keterangan	85
Gambar 5.3 Sequence Diagram Menyimpan Surat Keterangan	86
Gambar 5.4 Sequence Diagram Menghapus Surat Keterangan	86
Gambar 5.5 <i>Class diagram</i> untuk implementasi	87
Gambar 5.6 <i>Class diagram</i> sebagai domain <i>model</i>	88
Gambar 5.7 <i>Class diagram</i> dengan kerangka kerja Codeigniter	89
Gambar 5.8 <i>Physical Data Model</i> Sistem Informasi Pelayanan Surat Keterangan Desa Legundi	90
Gambar 5.9 Perancangan Antarmuka <i>Login</i>	102
Gambar 5.10 Perancangan Antarmuka Admin	102
Gambar 5.11 Perancangan Antarmuka Beranda	103
Gambar 5.12 Perancangan Antarmuka Kelola Bidang Kelahiran	103
Gambar 5.13 Perancangan Antarmuka Pengaturan	104
Gambar 6.1 Implementasi Antarmuka <i>Login</i>	112
Gambar 6.2 Implementasi Antarmuka Admin	112
Gambar 6.3 Implementasi Antarmuka Kelola Bidang Kelahiran	113
Gambar 6.4 Implementasi Antarmuka Kelola Pengguna	113

Gambar 6.5 Implementasi Antarmuka Pengaturan	114
Gambar 7.1 Kodifikasi Pengujian Validasi	115
Gambar 7.2 Hasil Pengujian Responsif Resolusi Layar 1366px × 768px 14 inci ..	121
Gambar 7.3 Hasil Pengujian Responsif Resolusi Layar 1080px × 1920px 5.5 inci	122
Gambar 7.4 Kategori <i>compatibility issues</i> pada SortSite	123
Gambar 7.5 Hasil Pengujian Kompatibilitas	123



BAB 1 PENDAHULUAN

1.1 Latar belakang

Pemenuhan kebutuhan pelayanan diatur dalam perundang-undangan. Kebutuhan pelayanan atas barang, jasa, atau pelayanan administratif disediakan oleh penyelenggara pelayanan publik. (Indonesia, 2009). Merupakan kewajiban negara dan amanat UUD Republik Indonesia, setiap warga negara dan penduduk mendapatkan pelayanan untuk kebutuhan dasar dan haknya.

Desa Legundi adalah desa yang berada pada wilayah Kecamatan Karangjati Kabupaten Ngawi. Dalam kasus Pemerintah Desa Legundi, pelayanan publik seperti pengurusan dokumen Surat Keterangan atau SK seperti contoh pengurusan SK Kematian, SK Kelahiran, SK KTP Proses dan lainnya masih menggunakan cara konvensional. Menurut wawancara penulis kepada perangkat desa cara konvensional yang dimaksud adalah dalam pengurusan dokumen surat keterangan tahapan yang harus dilalui adalah dengan menuliskan identitas yang mengajukan surat keterangan ke dalam buku registrasi. Selanjutnya, nomor dari buku registrasi digunakan untuk nomor surat. Setelah itu, identitas lainnya beserta nomor surat dimasukkan kedalam draf surat keterangan. Dalam kasus diatas terdapat berbagai masalah. Pertama, terkadang perangkat desa lupa untuk menulis identitas pada buku registrasi. Kedua, buku registrasi pencatatan surat keterangan rawan hilang atau rusak. Ketiga, terdapat dua kali kerja yaitu pencatatan pada buku registrasi dan pembuatan draft dokumen digital. Selain itu, menurut wawancara dengan 10 penduduk desa Legundi dihasilkan 9 orang mempunyai smartphone/perangkat komputer, 6 orang menyetujui apabila permohonan surat keterangan dapat dilakukan di tempat manapun dan dengan waktu yang tidak terikat. Dengan hasil wawancara tersebut dapat disimpulkan bahwa masyarakat Desa Legundi pada saat ini sudah mempunyai perangkat yang dapat membantu berjalannya sistem yang akan dibuat dan menyetujui atau mendukung sistem yang akan dibuat.

Melihat permasalahan yang terjadi pada pelayanan surat keterangan Pemerintah Desa Legundi, sangat diperlukan alat bantu yang memiliki fungsi untuk permohonan surat keterangan dan dokumentasi surat keterangan yang diharapkan dapat membantu dan mempermudah perangkat Desa Legundi serta penduduk Desa Legundi dalam permohonan surat keterangan. Oleh sebab itu, tujuan dari penelitian ini adalah untuk mengembangkan sebuah alat bantu berupa sistem informasi yang dapat melakukan fungsi-fungsi yang telah disebutkan diatas. Dalam permasalahan yang telah disebutkan diatas alat bantu harus bisa digunakan dari mana saja, mengatasi penyimpanan dan pembuatan permohonan surat keterangan dan mendukung jalannya proses bisnis instansi. Oleh sebab itu,

sistem informasi berbasis web menjadi pilihan untuk menyelesaikan masalah tersebut.

Menurut Sommerville (2011) pendekatan berorientasi objek dapat digunakan untuk mengembangkan sistem informasi karena lebih mudah menghadapi perubahan dengan pendekatan fungsional. Dalam pendekatan berorientasi objek menambahkan atau mengubah layanan pada suatu objek seharusnya tidak begitu berdampak pada objek lain. Pada pendekatan berorientasi objek lebih mudah dalam memetakan desain karena objek berisi data dan operasi yang merupakan visualisasi dari dunia nyata. Dalam bukunya Sommerville juga mengatakan untuk mengembangkan sistem berorientasi objek ada suatu proses pemodelan sistem. Untuk membangun suatu model abstrak dari sistem dengan pandangan yang berbeda untuk setiap model disebut proses pemodelan sistem. Pada proses pemodelan sistem untuk merepresentasikan sistem notasi grafis sebagai alat bantu. UML atau *unified modeling language* pada prakteknya telah menjadi standar untuk pemodelan berorientasi objek. UML digunakan untuk membantu proses analisis persyaratan, perancangan serta untuk dokumentasi struktur dan operasi sistem.

Menurut Zeaaraoui, et al. (2012) dengan melakukan pendekatan dengan OOA atau OOD dapat digunakan untuk mengatasi kelemahan dalam proses pengembangan perangkat lunak ketika pindah ke tahap konstruksi. Oleh karena itu pendekatan OOA atau OOD digunakan dalam penelitian ini. Untuk pengembangan sistem yang menerapkan pendekatan berorientasi objek digunakan bahasa pemrograman dan kerangka kerja yang mendukung pendekatan berorientasi objek. Menurut Sudarevic dan Vucsanovic (2011) kerangka kerja web menawarkan banyak keuntungan teknis dan organisasi (misalnya pengembangan lebih cepat dan struktur aplikasi yang lebih tertata) dibandingkan metode pengembangan klasik. Selain itu, pemrograman yang menggunakan kerangka kerja web lebih nyaman bagi pengembang, karena mereka tidak harus berurusan dengan banyaknya tugas pemrograman umum. CodeIgniter merupakan salah satu kerangka kerja yang mendukung pendekatan berorientasi objek dengan bahasa pemrograman PHP. Pada saat pengembangan sistem, komponen adalah penerapan dari objek yang dikenali pada desain. Komponen akan bertanggung jawab mengadakan fungsi yang merupakan kesanggupan sistem guna memenuhi persyaratan pengguna. Guna memastikan persyaratan pengguna telah dipenuhi oleh sistem maka kegiatan pengujian dapat dilakukan.

Berdasarkan penjelasan yang diuraikan sebelumnya, dan untuk mengembangkan sistem informasi pelayanan surat keterangan dengan mempraktikkan serangkaian kegiatan analisis, perancangan, dan implementasi dengan pendekatan berorientasi objek, dilakukan penelitian ini. Setelah itu, hasil dari pembangunan sistem informasi pelayanan surat keterangan diuji untuk menunjukkan hasil sistem informasi yang dikembangkan sesuai dengan

persyaratan yang telah dikenali guna mendukung pelayanan surat keterangan pada Pemerintah Desa Legundi.

1.2 Rumusan masalah

Berdasarkan pada masalah penelitian yang dijabarkan pada latar belakang maka secara umum dapat dirumuskan masalah sebagai berikut:

Mampukah dibuat sistem informasi yang mampu mendukung proses pelayanan publik pengurusan surat keterangan pada Pemerintah Desa Legundi?

Untuk dapat menjawab rumusan masalah umum diatas, dirumuskan permasalahan yang lebih khusus sebagai berikut:

1. Bagaimana hasil pemodelan proses bisnis dalam hal pengurusan surat keterangan Pemerintah Desa Legundi?
2. Bagaimana hasil analisis persyaratan Sistem Informasi Pelayanan Surat Keterangan dalam hal pengurusan surat keterangan Pemerintah Desa Legundi?
3. Bagaimana hasil perancangan Sistem Informasi Pelayanan Surat Keterangan dalam hal pengurusan surat keterangan Pemerintah Desa Legundi?
4. Bagaimana hasil implementasi Sistem Informasi Pelayanan Surat Keterangan dalam hal pengurusan surat keterangan Pemerintah Desa Legundi?
5. Bagaimana hasil pengujian dari Sistem Informasi Pelayanan Surat Keterangan dalam hal pengurusan surat keterangan Pemerintah Desa Legundi?

1.3 Batasan Masalah

Pada penelitian ini penulis membatasi masalah sebagai berikut:

1. Sistem Informasi Pelayanan Publik Pemerintah Desa Legundi mencakup pelayanan Surat Keterangan Kematian, Surat Keterangan Kutipan Kelahiran, Surat Keterangan Kelahiran, Surat Keterangan KTP Proses, Surat Keterangan Belum Menikah.
2. Sistem Pelayanan Publik diimplementasikan berbasis web.
3. Objek penelitian adalah Pemerintah Desa Legundi.

1.4 Tujuan Penelitian

Tujuan umum dari penelitian ini adalah:

Membangun atau membuat Sistem Informasi Pelayanan Surat Keterangan yang mampu mendukung proses permohonan surat keterangan.

Tujuan khusus dari penelitian ini adalah:

1. Memodelkan proses bisnis.
2. Menganalisis persyaratan Sistem Informasi Pelayanan Surat Keterangan.
3. Merancang Sistem Informasi Pelayanan Surat Keterangan.

4. Mengimplementasi Sistem Informasi Pelayanan Surat Keterangan.
5. Menguji Sistem Informasi Pelayanan Surat Keterangan.

1.5 Manfaat

Penelitian ini diharapkan mampu mempermudah dan mempercepat kinerja dari perangkat Pemerintah Desa Legundi dalam melakukan pembuatan surat keterangan serta mempermudah dan mempercepat permohonan surat keterangan yang dilakukan oleh penduduk Desa Legundi.

1.6 Sistematika Penulisan

Sistematika penulisan penelitian ini adalah sebagai berikut:

BAB I. Pendahuluan

Pada bab I terdapat latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian dan sistematika penulisan.

BAB II. Landasan Kepustakaan

Landasan kepustakaan berisi dasar teori serta teori pendukung yang digunakan sebagai rujukan mengerjakan penelitian dan merupakan inti dari dasar teori yang akan digunakan dalam penelitian.

BAB III. Metodologi

Metodologi berisi metode penelitian yang dipakai saat melakukan penelitian. Menguraikan mengenai objek penelitian, variabel, metode penelitian, metode pengumpulan data, metode perancangan, metode implementasi serta metode pengujian.

BAB IV. Pemodelan Proses Bisnis dan Analisis Persyaratan Sistem

Membahas tentang pemodelan proses bisnis *as-is* dan *to-be* dan analisis persyaratan Sistem Informasi Pelayanan Surat Keterangan berbasis web untuk memaksimalkan kinerja Pemerintah Desa Legundi yang terdiri dari analisis masalah dan identifikasi tipe pemangku kepentingan, kebutuhan pemangku kepentingan dan pengguna, pengguna, fitur serta pemodelan *use case*.

BAB V. Perancangan

Terdiri dari perancangan sistem dalam diagram hubungan antar objek, hubungan antar kelas, perancangan basis data, perancangan algoritme serta antarmuka pengguna yang

dibutuhkan sebagai syarat perancangan. *Sequence diagram* dan diagram kelas akan dijelaskan dalam bab perancangan.

BAB VI. Implementasi

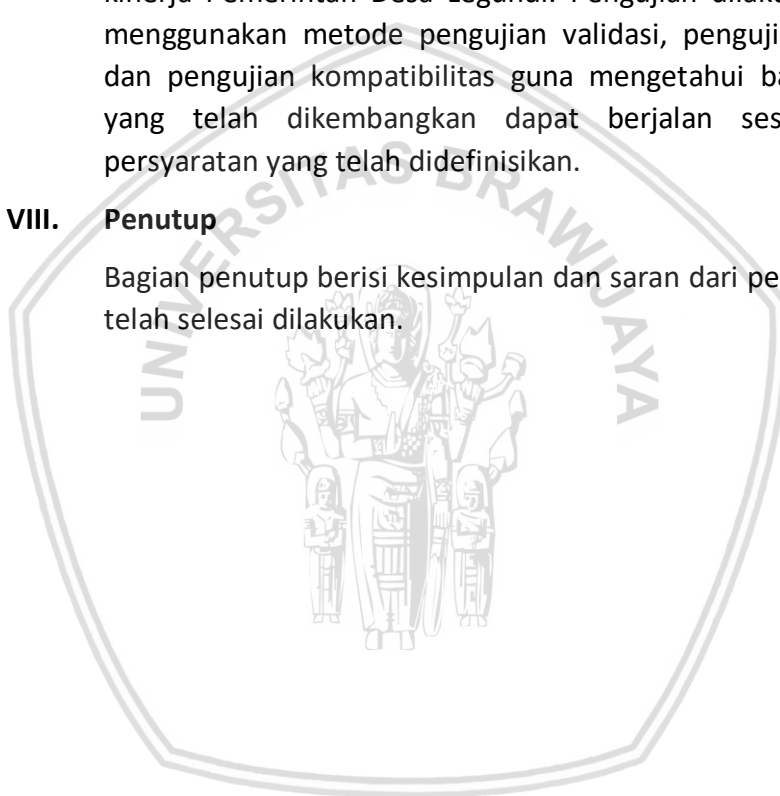
Membahas mengenai implementasi yang dilakukan dari sistem seperti spesifikasi lingkungan implementasi, hasil implementasi algoritme serta hasil implementasi antarmuka pengguna.

BAB VII. Pengujian

Memuat proses dan hasil dari pengujian Sistem Informasi Pelayanan Surat Keterangan berbasis web untuk memaksimalkan kinerja Pemerintah Desa Legundi. Pengujian dilakukan dengan menggunakan metode pengujian validasi, pengujian responsif dan pengujian kompatibilitas guna mengetahui bahwa sistem yang telah dikembangkan dapat berjalan sesuai dengan persyaratan yang telah didefinisikan.

BAB VIII. Penutup

Bagian penutup berisi kesimpulan dan saran dari penelitian yang telah selesai dilakukan.



BAB 2 LANDASAN KEPUSTAKAAN

2.1 Kajian Pustaka

Pada bagian ini terdapat teori-teori atau temuan hasil penelitian terdahulu yang relevan dengan permasalahan yang diambil. Teori-teori atau temuan hasil penelitian terdahulu digunakan sebagai acuan atau dasar yang dapat digunakan sebagai data pendukung. Pada Tabel 2.1 terdapat beberapa hasil penelitian terdahulu.

Tabel 2.1 Penelitian Terdahulu

No	Peneliti	Judul	Hasil/Temuan
1	(Zeaaraoui, et al., 2012)	<i>Object-oriented Analysis and Design Approach for Requirements Engineering</i>	Rekayasa kebutuhan layak mendapatkan perhatian yang lebih kuat. Banyak pendekatan dikembangkan untuk memodelkan persyaratan, tetapi semua menunjukkan kelemahan dalam proses pengembangan perangkat lunak ketika pindah ke tahap konstruksi. Dalam penelitian ini penulis merepresentasikan metode yang digunakan untuk menyusun dan memodelkan rekayasa kebutuhan dengan OOA atau OOD. Bahwa dengan melakukan pendekatan dengan OOA atau OOD dapat mengatasi masalah diatas. Hasil penelitian tersebut yang melandasi penggunaan OO.
2	(Sudarevic & Vucsanovic, 2011)	<i>Use of Web Application Frameworks in the Development of Small Applications</i>	Kerangka kerja web menawarkan banyak keuntungan teknis dan organisasi (misalnya pengembangan lebih cepat dan struktur aplikasi yang lebih tertata) dibandingkan metode pengembangan klasik. Selain itu, pemrograman yang menggunakan kerangka kerja web lebih nyaman bagi pengembang, karena mereka tidak harus berurusan dengan banyaknya task pemrograman umum. Terlepas dari keunggulan di atas, pengembangan aplikasi yang lebih kecil biasanya dilakukan tanpa kerangka kerja web karena kerugian yang dirasakan.

No	Peneliti	Judul	Hasil/Temuan
			Hasil penelitian tersebut yang melandasi penggunaan kerangka kerja web.
3	(Pamungkas, et al., 2018)	Pengembangan Sistem Informasi Persediaan Barang, Harga Pokok Produksi, dan Transaksi Penjualan Berbasis Web (Studi Pada Screen Printing Sidoarjo)	Penelitian ini menggunakan metode <i>waterfall</i> dan berorientasi objek menggunakan kerangka kerja web codeigniter. Menghasilkan 13 <i>sequence diagram</i> , 5 <i>class model</i> , 6 <i>class controller</i> , 4 tabel perancangan basis data, 7 perancangan antarmuka. Pada pengujian validasi menghasilkan 100% valid pada 23 kasus uji. Pada evaluasi waktu kinerja sistem menghasilkan 128,45 detik lebih cepat dari metode manual. Dari kesimpulan diatas metode <i>waterfall</i> masih relevan untuk dipakai untuk pengembangan sistem informasi disandingkan dengan kerangka kerja web codeigniter.

2.2 Pelayanan Publik

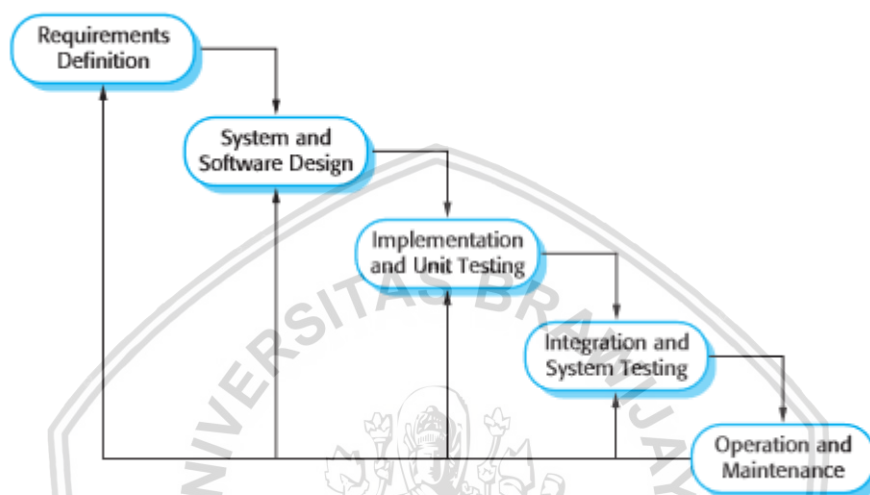
Pelayanan publik adalah kegiatan atau rangkaian kegiatan dalam rangka pemenuhan kebutuhan pelayanan sesuai dengan peraturan perundang-undangan bagi setiap warga negara dan penduduk atas barang, jasa, dan/atau pelayanan administratif yang disediakan oleh penyelenggara pelayanan publik (Indonesia, 2009). Pelayanan publik menjadi tanggung jawab dan dilaksanakan oleh Instansi Pemerintah, Badan Usaha Milik Negara, Badan Usaha Milik Daerah, dll. Selain instansi pemerintahan, pelayanan publik dapat juga diselenggarakan oleh pihak swasta, misal rumah sakit swasta, perusahaan angkutan milik swasta.

2.3 Pemerintah Desa Legundi

Pemerintah Desa Legundi adalah lembaga pemerintah yang bertugas mengelola wilayah tingkat Desa Legundi Kecamatan Karangjati Kabupaten Ngawi Jawa Timur. Lembaga ini diatur dalam Peraturan Pemerintah No. 72 Tahun 2005 tentang pemerintahan desa yang diterbitkan untuk melaksanakan ketentuan pasal 216 ayat (1) Undang Undang Nomor 32 Tahun 2004 tentang pemerintahan desa. Peraturan mengenai desa lebih lanjutnya ditetapkan dalam Perda yang berpedoman pada Peraturan Pemerintah. Pemerintah Desa juga wajib mengakui dan menghormati hak, asal-usul dan adat istiadat desa.

2.4 Pengembangan Perangkat Lunak dengan Model *Waterfall*

Model *waterfall* merupakan proses fundamental yang terdiri dari tahap spesifikasi, pengembangan, validasi, dan evolusi. Setiap tahap direpresentasikan sebagai proses yang terpisah seperti spesifikasi kebutuhan, perancangan perangkat lunak, implementasi, pengujian, dan sebagainya. Model *waterfall* diawali dengan proses perancangan secara terencana yang berguna sebagai dasar pengembangan berikutnya (Sommerville, 2011).



Gambar 2.1 Model *Waterfall*

Sumber : Sommerville (2011)

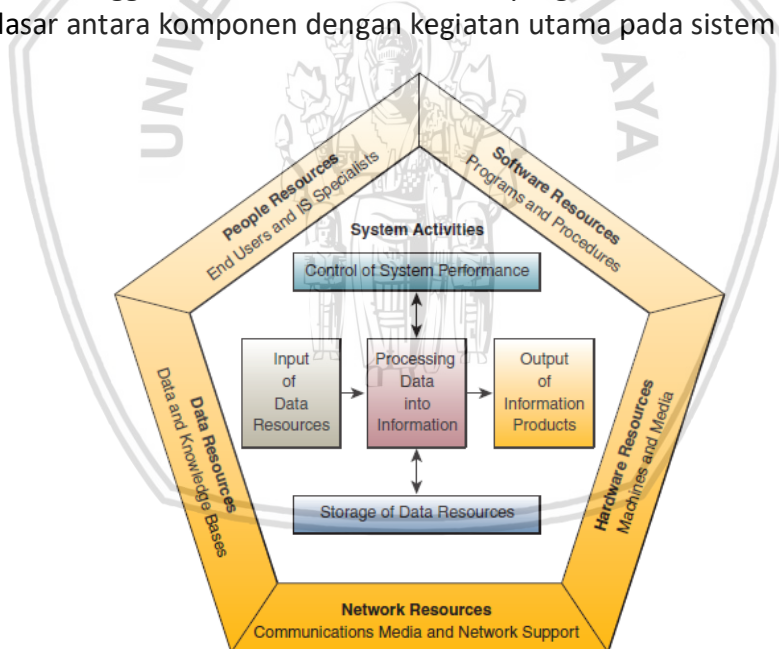
Pada Gambar 2.1 dapat dilihat metode pengembangan model *waterfall*. Tahapan utama dari model *waterfall* secara langsung menerapkan aktivitas dasar dalam pengembangan:

1. Analisis dan definisi kebutuhan. Proses analisis merupakan penentuan layanan, batasan, dan tujuan berdasarkan konsultasi dengan pengguna sistem. Selanjutnya didefinisikan secara rinci sebagai spesifikasi sistem.
2. Perancangan sistem dan perangkat lunak. Proses perancangan mengalokasikan kebutuhan pada perangkat keras maupun perangkat lunak dengan membangun arsitektur sistem secara keseluruhan.
3. Implementasi dan Pengujian Unit. Proses implementasi merealisasikan rancangan sebagai sebuah program maupun unit program. Pengujian unit dilakukan untuk memverifikasi setiap unit telah memenuhi spesifikasi yang didefinisikan di awal.
4. Integrasi dan Pengujian Sistem. Proses integrasi dilakukan dengan menggabungkan unit-unit program menjadi sebuah sistem yang utuh. Selanjutnya sistem akan diuji secara keseluruhan kepada pengguna sistem.

5. Operasional dan Pemeliharaan. Tahap ini merupakan tahap dimana sistem mulai digunakan. Selanjutnya tahap pemeliharaan berupa perawatan perangkat lunak, pengecekan kesalahan dan perbaikan kekurangan pada sistem. Pada tahap ini memungkinkan terjadinya perulangan menuju tahap sebelumnya.

2.5 Sistem Informasi

Sistem dapat didefinisikan sebagai sekumpulan komponen yang berhubungan, mempunyai batasan dan berkolaborasi demi sebuah tujuan. Secara umum, sistem informasi dijelaskan sebagai perpaduan antara sumber data, manusia, perangkat keras, perangkat lunak, jaringan komunikasi, prosedur serta kebijakan dalam pengelolaan dan pendistribusian informasi di organisasi. Sistem informasi dimanfaatkan manusia sebagai sarana komunikasi melalui perangkat keras, perangkat lunak, jaringan, dan data (Marakas & O'Brian, 2013). Sistem informasi adalah sebuah sistem yang berfungsi untuk memperoleh data sebagai sebuah masukan dan memproses masukan data untuk menghasilkan keluaran sebuah informasi yang berguna untuk pengguna sehingga dapat mencapai tujuan. Gambar 2.2 menggambarkan sistem informasi yang dimodelkan sebagai sebuah konsep dasar antara komponen dengan kegiatan utama pada sistem informasi.



Gambar 2.2 Sistem Informasi

Sumber: Marakas & O'Brian (2013)

2.6 Proses Bisnis

Proses bisnis dijelaskan sebagai sekumpulan kegiatan yang setidaknya memiliki satu masukan atau lebih dan memberikan hasil keluaran yang bernilai bagi pelanggan (Monk & Wagner, 2013). Dalam hal ini, pelanggan menurut Monk dan Wagner dibedakan menjadi pelanggan eksternal dan pelanggan

internal. Pelanggan eksternal digambarkan sebagai seorang maupun kelompok yang melakukan pembelian produk. Sedangkan pelanggan internal adalah orang ataupun kelompok pada perusahaan dari berbagai departemen. Menurut Becker, et al. sekumpulan kegiatan pada sebuah perusahaan terdiri dari proses inti dan proses pendukung. Proses inti didefinisikan sebagai proses kegiatan langsung yang berkaitan dengan produk perusahaan, proses inti berperan penting dalam menghasilkan keuntungan bagi perusahaan. Sebaliknya, proses pendukung merupakan sekumpulan kegiatan yang tidak berkaitan dalam menghasilkan nilai bagi pelanggan, tetapi berperan penting dalam melakukan proses inti pada perusahaan. Terdapat dua pendekatan untuk dapat mengetahui proses bisnis suatu perusahaan, yaitu melakukan proses analisis dan memodelkan proses bisnis dalam bentuk *as-is* dan *to-be*.

2.6.1 Analisis dan Pemodelan Bisnis As-Is

Menurut Becker, et al. analisis dan pemodelan proses bisnis *as-is* merupakan proses pengumpulan dan pemodelan proses bisnis yang sedang berjalan pada perusahaan dengan tujuan memberikan informasi terkait keadaan aktual dan mengevaluasi kekurangan dari proses bisnis yang ada untuk merekomendasikan perbaikan proses bisnis pada perusahaan. Hal yang dapat dilakukan untuk mengidentifikasi proses bisnis *as-is* salah satunya dengan melakukan wawancara dengan pekerja yang terlibat langsung dalam operasional proses bisnis. Cara lainnya dapat dilakukan melalui metode observasi yaitu dengan cara melihat dan memantau langsung pekerja dalam melakukan pekerjaannya pada perusahaan (Przybylek, 2011). Menurut Przybylek, hasil dari identifikasi dapat digunakan sebagai gambaran diagram untuk memodelkan proses bisnis *as-is*. Diagram meliputi proses bisnis fundamental yang ada di perusahaan dan setiap kegiatan dalam proses bisnis harus dimodelkan. Kegiatan yang kompleks dapat dimodelkan dengan sub-proses dari proses yang umum. Setelah analisis dan pemodelan proses bisnis *as-is* selesai maka dibutuhkan persetujuan pemangku kepentingan untuk memastikan bahwa proses bisnis yang dimodelkan telah sesuai.

2.6.2 Analisis dan Pemodelan Proses Bisnis To-Be

Menurut Becker, et al. pemodelan dan analisis proses bisnis *to-be* adalah kegiatan atau serangkaian kegiatan untuk memodelkan proses bisnis yang telah mengalami peningkatan. Pemodelan proses bisnis *to-be* dilakukan berdasarkan analisis terhadap proses bisnis *as-is*. Pemindahan atau migrasi dari proses bisnis *as-is* menjadi proses bisnis *to-be* dapat dilakukan secara bertahap. Penerapan proses bisnis *to-be* dapat dilakukan dalam jangka waktu menengah atau dalam jangka panjang. Pemangku kepentingan terlibat dalam analisis dan pemodelan proses bisnis *to-be*. Keterlibatan pemangku kepentingan dalam analisis dan pemodelan proses bisnis *to-be* mempunyai tujuan untuk dapat memperoleh kemungkinan peningkatan proses bisnis pada organisasi atau perusahaan (Przybylek, 2011). Przybylek juga menjelaskan, penerapan otomasi pada beberapa proses bisnis mungkin dilakukan. Akan tetapi penerapan otomasi tidak menjamin

terjadinya peningkatan yang signifikan. Oleh karena itu, keterlibatan pemangku kepentingan dalam pemodelan proses bisnis *to-be* sangatlah penting.

2.7 User Story

User Story adalah deskripsi singkat dan sederhana tentang fitur yang diceritakan dari perspektif orang yang menginginkan kemampuan baru, biasanya pengguna atau pelanggan dari sistem. Mereka biasanya menggunakan contoh sederhana seperti:

Sebagai <jenis pengguna>, saya ingin <beberapa tujuan> sehingga <beberapa alasan> (Cohn, 2008).

Beberapa keuntungan *user story* yang pertama adalah *user story* menekankan komunikasi verbal. Bahasa tertulis seringkali tidak tepat dan tidak ada jaminan bahwa pelanggan dan pengembang akan menafsirkan pernyataan dengan cara yang sama. Yang kedua adalah dapat digunakan dengan mudah untuk perencanaan proyek. *User story* pengguna ditulis dalam dokumen sehingga dapat diperkirakan seberapa sulit dan seberapa memakan waktu yang akan digunakan. Yang ketiga adalah *user story* dapat digunakan untuk mempermudah merinci kebutuhan pengguna (Cohn, 2004).

2.8 Business Process Modeling Notation (BPMN)

Tujuan dari BPMN adalah untuk menstandarisasi model proses bisnis dan notasi dalam menghadapi berbagai macam notasi pemodelan dan sudut pandang yang berbeda. Dengan demikian BPMN akan menyediakan cara sederhana untuk komunikasi proses informasi ke pengguna bisnis lain, pelaksana proses, pelanggan, dan pemasok. BPMN terdiri dari elemen-elemen Kolaborasi, diagram proses elemen, termasuk semua tipe *task*, sub proses, *CallActivity*, semua jenis *gateway*, semua jenis *event*, dll. Tujuan utama dari BPMN adalah memberikan notasi yang mudah atau gampang dimengerti oleh semua pengguna bisnis, dari analis bisnis yang membuat semua draf awal dari proses, kepada pengembang yang bertanggung jawab atas teknologi yang akan melakukan proses tersebut, dan yang terakhir untuk orang yang akan mengelola dan memantau proses tersebut. Dengan demikian BPMN menciptakan standar untuk kesenjangan antara proses desain dan pengembangan proses bisnis (Group, 2011). Dalam *Object Management Group* juga menambahkan bahwa BPMN menyediakan sekumpulan notasi yang dibagi menjadi 5 kategori. Kelima kategori tersebut adalah *flow objects*, *data*, *connecting objects*, *swimlanes*, *artifacts*.

2.8.1 Flow Object

Notasi ini menjelaskan mengenai *behavior* atau perilaku yang dimiliki oleh proses bisnis. Terdapat 3 elemen dalam kategori ini. Tiga elemen tersebut terdiri dari:

1. Events

Definisi dari *events* adalah notasi memicu terhadap alur aktivitas di dalam model proses bisnis. Notasi *event* terdiri dari 3 tipe. Tiga tipe tersebut adalah *start event*, *intermediate event*, dan *end*. Tabel 2.2 menjelaskan mengenai notasi dari *event*.

Tabel 2.2 Notasi Event

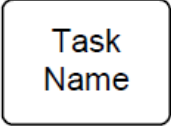
Notasi	Nama	Deskripsi
	<i>Start Event</i>	<i>Start Event</i> menggambarkan atau mewakili permulaan sebuah proses.
	<i>Intermediate Event</i>	<i>Intermediate Event</i> berada diantara <i>start event</i> dan <i>end</i> . <i>Intermediate Event</i> mempengaruhi alur proses bisnis. <i>Intermediate Event</i> digunakan untuk memulai dan/atau mengakhiri sebuah proses secara langsung. <i>Intermediate Event</i> terdiri dari berbagai macam jenis untuk mewakili event yang berbeda-beda. Setiap jenis diindikasikan dengan simbol yang berbeda.
	<i>End</i>	<i>End</i> mewakili atau menggambarkan akhir sebuah proses.

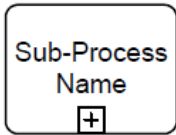
Sumber: Object Management Group (2011)

2. Activity

Definisi dari *activity* adalah notasi yang mewakili atau menggambarkan mengenai sebuah pekerjaan atau *task* dalam sebuah proses. Notasi *activity* terdiri dari 2 tipe. Dua tipe tersebut adalah *sub-process* dan *task*. Tabel 2.3 menjelaskan mengenai notasi dari *activity*.

Tabel 2.3 Notasi Activity

Notasi	Nama	Deskripsi
	<i>Task</i>	<i>Task</i> menggambarkan atau mewakili aktivitas yang bersifat tidak bisa dipecah ke tingkat yang lebih kecil.






Notasi	Nama	Deskripsi
	<i>Sub-process</i>	<i>Sub-process</i> menggambarkan atau mewakili sekumpulan aktivitas. <i>Sub-process</i> memiliki sekumpulan aktivitas didalamnya.

Sumber: Object Management Group (2011)

3. Gateways

Notasi *gateways* menjelaskan alur proses bisnis. Tabel 2.4 menjelaskan notasi *gateways*.

Tabel 2.4 Notasi Gateway






Notasi	Nama	Deskripsi
	<i>Exclusive</i>	<i>Exclusive</i> menjelaskan dari beberapa alur alternatif yang dibuat, hanya 1 alur alternatif yang dapat dijalankan sesuai dengan kondisi yang terpenuhi.
	<i>Parallel</i>	Untuk membuat alur paralel dan menyelaraskan alur paralel yang telah dibuat.
	<i>Event-based</i>	Mewakili percabangan yang alur alternatifnya berdasarkan sebuah <i>events</i> tertentu yang terjadi dalam proses bisnis.
	<i>Complex</i>	Dapat digunakan untuk memodelkan alur kompleks dalam proses bisnis.
	<i>Inclusive</i>	Mewakili percabangan yang alternatifnya berdasarkan kondisi didalamnya. Kondisi pada setiap alur diperiksa, kemudian alur yang memenuhi kondisi dijalankan. Dapat berjalan secara paralel sesuai dengan kondisi yang terpenuhi.

Sumber: Object Management Group (2011)

2.8.2 Data

Data digunakan untuk memvisualisasikan cara dokumen, data, dan objek lain digunakan dan diperbarui pada sebuah alur proses (IBM Knowledge Center, 2014). Kategori ini direpresentasikan oleh 4 notasi yang terdiri dari *data objects*, *data inputs*, *data outputs*, *data stores*. Masing-masing notasi dijelaskan pada Tabel 2.5.

Tabel 2.5 Notasi Data


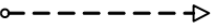


Notasi	Nama	Deskripsi
	<i>Data Object</i>	Menggambarkan informasi mengenai data atau dokumen yang diperlukan <i>activity</i> .
	<i>Data Object (Collection)</i>	Digunakan untuk menggambarkan <i>data object</i> yang berjumlah lebih dari 1.
	<i>Data Input</i>	Menggambarkan data atau dokumen atau objek lain yang menjadi dibutuhkan atau menjadi masukan sebuah atau serangkaian aktivitas.
	<i>Data Output</i>	Menggambarkan data atau dokumen atau objek lain yang menjadi menjadi keluaran sebuah atau serangkaian aktivitas.
	<i>Data Store</i>	Menggambarkan sebuah lokasi penyimpanan pembaruan data.

Sumber: Object Management Group (2011)

2.8.3 Connecting Object

Notasi *connecting objects* digunakan untuk menghubungkan notasi *flow object* dengan notasi-notasi BPMN lainnya. Tabel 2.6 menjelaskan beberapa notasi *connecting objects*.

Tabel 2.6 Notasi *Connecting Objects*



Notasi	Nama	Deskripsi
	<i>Sequence Flow</i>	Menggambarkan urutan aktivitas yang dijalankan dalam proses bisnis.
	<i>Message Flow</i>	Menggambarkan atau mewakili aliran yang dikirim dan diterima sebuah aktivitas yang dimiliki partisipan yang berbeda.
	<i>Associations</i>	Menggambarkan atau mewakili hubungan notasi <i>artifacts</i> yang berisi informasi dengan notasi BPMN lainnya.
	<i>Data Associations</i>	Menggambarkan atau mewakili hubungan aktivitas BPMN dengan notasi data.

Sumber: Object Management Group (2011)

2.8.4 Swimlanes

Swimlanes adalah wadah gambar untuk memisahkan atau mempartisi satu set aktivitas dari aktivitas lain. BPMN memiliki 2 tipe *swimlanes* yaitu *pool* dan *lane*. Tabel 2.7 menjelaskan notasi pada *swimlanes*.

Tabel 2.7 Notasi Swimlanes


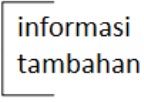
Notasi	Nama	Deskripsi
	<i>Pool</i>	Mewakili partisipan dalam sebuah proses bisnis. <i>Pool</i> adalah sebuah wadah untuk memisahkan proses dari <i>Pool</i> atau peserta lain.
	<i>Lane</i>	Sebuah partisi atau pemisah untuk mengkategorikan aktivitas dalam <i>pool</i> . <i>Lane</i> mewakili unit-unit yang dimiliki oleh sebuah partisipan proses bisnis.

Sumber: Object Management Group (2011)

2.8.5 Artifacts

Notasi-notasi *Artifacts* adalah objek grafis yang menyediakan informasi pendukung mengenai proses atau elemen dalam proses dan tidak mempengaruhi alur proses. Tabel 2.8 menjelaskan beberapa notasi *artifacts*.

Tabel 2.8 Notasi Artifacts

Notasi	Nama	Deskripsi
	<i>Group</i>	Untuk menggabungkan elemen grafis yang mempunyai kategori sama. <i>Group</i> tidak mempengaruhi alur yang didalam <i>group</i> .
	<i>Text Annotation</i>	Mekanisme untuk sebuah model untuk menyediakan sebuah teks tambahan kepada pembaca BPMN.

Sumber: Object Management Group (2011)

2.9 Pemodelan Use Case

Pemodelan *use case* memberikan pandangan dan pemahaman menyeluruh mengenai sistem yang akan dikembangkan dengan memberikan gambaran lengkap tentang sistem yang akan dibangun dan motivasi dibaliknya. Pemodelan *use case* juga membantu untuk memberikan gambaran yang ringkas dari sistem yang akan dibangun. Pemodelan *use case* juga menggambarkan cara pengguna

menggunakan sistem serta yang dilakukan sistem untuk penggunaanya (Bittner & Spence, 2002).

2.9.1 Identifikasi Tipe Pemangku Kepentingan

Sebelum memulai memodelkan *use case* atau persyaratan lainnya, sangat penting untuk mengerti pemangku kepentingan. Pemangku kepentingan adalah seorang individu yang secara material dipengaruhi oleh hasil dari sistem atau proyek yang menghasilkan sistem (Bittner & Spence, 2002). Langkah pertama yang harus dilakukan untuk mengerti kumpulan pemangku kepentingan adalah mengidentifikasi tipe dari pemangku kepentingan yang dipengaruhi oleh sistem. Tipe pemangku kepentingan adalah klasifikasi seperangkat pemangku kepentingan dengan karakteristik dan hubungan yang sama dengan sistem dan/atau proyek yang akan menghasilkan sistem. Pemangku kepentingan biasanya dipecah menjadi kategori berikut:

1. Pengguna: Orang-orang yang akan mengambil peran dan didefinisikan sebagai aktor dalam model *use case*.
2. Sponsor: Para manager bisnis, pemodal, pemegang saham, dll yang berinvestasi dalam produksi sistem.
3. Pengembang: Manager proyek, pengelola sistem, penguji, dll yang terlibat dalam produksi dan mendukung sistem.
4. Pihak yang berwenang: Para ahli dalam aspek tertentu. Termasuk otoritas legislatif, departement tata kelola organisasi, badan pengatur eksternal dan internal, pakar teknologi, dll.
5. Pelanggan: Orang-orang dan/atau organisasi yang sebenarnya akan membeli sistem final.

Akan tetapi kategori pemangku kepentingan tidak terbatas pada apa yang telah disebutkan diatas. Tipe pemangku kepentingan bisa saja dikembangkan mengikuti kebutuhan yang diperlukan.

2.9.2 Analisis Masalah

Analisis masalah didefinisikan sebagai proses memahami masalah yang ada pada dunia nyata dan kebutuhan pengguna dan mengusulkan solusi untuk mengatasi kebutuhan tersebut. Tujuan dari analisis masalah adalah untuk mendapatkan pemahaman yang lebih baik sebelum pengembangan dimulai, dari masalah yang sedang dipecahkan (Leffingwell & Widrig, 2003). Cara terbaik untuk menangkap masalah adalah dengan membuat pernyataan masalah. Pernyataan masalah adalah ringkasan netral dari pemahaman bersama para pemangku kepentingan tentang masalah yang harus dipecahkan. Pernyataan masalah termasuk penilaian dampak masalah dan manfaat yang akan diperoleh oleh

solusinya (Bittner & Spence, 2002). Tabel 2.9 menunjukkan cara mendokumentasikan pernyataan masalah.

Tabel 2.9 Kerangka Dokumentasi Pernyataan Masalah

<i>The problem of</i>	Berisi deskripsi masalah.
<i>Affects</i>	Berisi pemangku kepentingan yang terpengaruh oleh masalah.
<i>The impact of which is</i>	Berisi deskripsi dampak masalah terhadap pemangku kepentingan dan kegiatan bisnis.
<i>A successful solution would</i>	Berisi beberapa solusi dan manfaat utama yang ditawarkan untuk menyelesaikan masalah.

Sumber: Bittner & Spence (2002)

2.9.3 Identifikasi Kebutuhan Pemangku Kepentingan dan Pengguna

Refleksi masalah bisnis, pribadi maupun operasional atau peluang yang harus ditunjukkan untuk membenarkan pertimbangan, pembelian atau penggunaan sistem baru (Leffingwell & Widrig, 2003). Mengambil kebutuhan pemangku kepentingan memungkinkan untuk memahami cara dan sejauh mana aspek-aspek berbeda dari masalah tersebut mempengaruhi berbagai jenis pemangku kepentingan. Menangkap atau memperoleh kebutuhan pemangku kepentingan melengkapi dan memberikan pemahaman yang lebih dalam pernyataan masalah yang telah didefinisikan. Kebutuhan pemangku kepentingan dapat dianggap sebagai ekspresi dari persyaratan bisnis yang sebenarnya dari sistem. Pemangku kepentingan akan memberikan wawasan tentang akar dari masalah secara keseluruhan dan menentukan satu set pernyataan solusi persyaratan independen yang apabila terpenuhi akan memecahkan masalah bisnis yang mendasarinya (Bittner & Spence, 2002). Leffingwell dan Widrig juga menjelaskan dalam bukunya untuk mengerti kebutuhan pemangku kepentingan dapat menggunakan teknik sebagai berikut:

1. Wawancara dan kuisisioner
2. *Requirement workshop*
3. *Brainstorming sessions and Idea Reduction*
4. *Storyboarding*

Pemilihan teknik harus didasarkan pada tipe pemangku kepentingan, keahlian yang dimiliki oleh tim, skala masalah, dll.

2.9.4 Identifikasi Fitur

Fitur adalah cara mudah untuk menggambarkan fungsionalitas tanpa harus menjelaskannya secara terperinci. Dalam kasus apapun fitur menjadi cara yang

berguna dan mudah untuk menggambarkan fungsionalitas sistem baru tanpa harus menjelaskannya secara terperinci. Fitur didefinisikan sebagai layanan yang disediakan sistem untuk memenuhi satu atau lebih kebutuhan pemangku kepentingan (Leffingwell & Widrig, 2003). Fitur adalah kemampuan tingkat tinggi layanan atau kualitas dari sistem yang diperlukan untuk memberi manfaat kepada pengguna dan membantu memenuhi kebutuhan pemangku kepentingan serta pengguna. Sebuah fitur juga memberikan ringkasan manfaat dari produk yang akan dibangun. (Bittner & Spence, 2002). Bittner juga menjelaskan untuk memprioritaskan fitur dengan aturan MoSCoW. MoSCoW (*Must Have, Should Have, Could Have, Won't Have this time*) digunakan untuk memprioritaskan persyaratan, meskipun teknik ini juga banyak berguna di bidang lainnya. Direkomendasikan tidak lebih dari 60% upaya untuk Must Have untuk sebuah project, 40% untuk Shoulds dan Could. Apapun yang lebih tinggi dari 60% menimbulkan resiko terhadap keberhasilan proyek, kecuali lingkungan dipahami dengan baik, tim yang baik serta resiko eksternal diminimalisir (Handbook, 2008). Tabel 2.10 adalah contoh cara mendokumentasikan fitur.

Tabel 2.10 Contoh Tabel Fitur

Kode Fitur		Deskripsi
Kode fitur	identitas	Penjelasan tentang yang dapat dilakukan oleh sistem

Sumber: Diadaptasi dari Bittner & Spence (2002)

2.9.5 Identifikasi Persyaratan Fungsional dan Persyaratan Nonfungsional

Bittner dan Spence (2002) dalam bukunya mengatakan, persyaratan fungsional adalah aksi atau tindakan yang harus dapat dilakukan oleh sistem tanpa mempertimbangkan kendala fisik. Persyaratan fungsional menentukan perilaku masukan dan keluaran dari suatu sistem. Persyaratan non-fungsional adalah persyaratan yang menentukan kualitas lain yang harus dimiliki sistem, seperti *usability*, keandalan, performa dari sistem dan dukungan sistem. Banyak persyaratan yang tidak berfungsi dan hanya mendeskripsikan atribut sistem atau atribut lingkungan sistem.

Persyaratan fungsional adalah pernyataan layanan yang seharusnya dilakukan oleh sistem, cara sistem bereaksi terhadap masukan tertentu serta cara sistem berperilaku dalam situasi tertentu. Persyaratan non-fungsional adalah batasan pada layanan atau fungsi yang ditawarkan oleh sistem. Persyaratan non-fungsional sering berlaku untuk sistem secara keseluruhan (Sommerville, 2011).

Persyaratan non-fungsional menjelaskan cara sistem harus berperilaku ketika disajikan dengan masukan atau kondisi tertentu. Persyaratan non-fungsional digunakan untuk mengungkapkan beberapa atribut sistem atau atribut dari lingkungan sistem (Leffingwell & Widrig, 2003).

2.9.6 Pemodelan *Use Case*

Model *use case* adalah himpunan semua *use case*, aktor, hubungan antar aktor dan *use case* yang digunakan untuk menggambarkan sistem tertentu. UML mendefinisikan model tertutup dari sistem subjek. Dengan kata lain model *use case* adalah deskripsi lengkap dari suatu sistem dari perspektif tertentu. Dalam pengembangan perangkat lunak atau sistem, seperti dalam banyak bidang lainnya, model adalah penyerdehanaan realitas, dibuat untuk meningkatkan pemahaman tentang sistem yang sedang dibangun. Model *use case* menyajikan sistem dalam hal penggunaannya. Model *use case* menggambarkan semua kemungkinan cara menggunakan sistem (Bittner & Spence, 2002). Aktor adalah sebuah peran yang mana pengguna dapat menggunakannya ketika berinteraksi dengan sistem. Aktor dapat mewakili individu atau seseorang atau sebuah sistem lain. *Use case* dideskripsikan cara aktor dan sistem untuk mencapai sebuah tujuan tertentu dan cara sistem kepada aktor untuk mencapai tujuan tertentu. Menceritakan cara sistem dan aktor berkolaborasi untuk memberi sebuah nilai. Model *use case* harus didukung dengan persyaratan lainnya karena model *use case* tidak menyediakan penjelasan atau deskripsi yang lengkap. Dalam model *use case* terdapat spesifikasi *use case*. Spesifikasi *use case* berisi penjelasan cara aktor dan sistem berkolaborasi atau bekerja sama. Tabel 2.11 menjelaskan format dokumentasi spesifikasi *use case*.

Tabel 2.11 Format Dokumentasi Spesifikasi *Use Case*

Brief Description	Deskripsi atau penjelasan singkat tentang <i>use case</i> .
Aktor	Aktor atau pengguna yang berinteraksi dengan <i>use case</i> .
Pre-condition	Kondisi yang harus dipenuhi sebelum <i>use case</i> dijalankan.
Post-condition	Kondisi yang dipenuhi atau kondisi yang terjadi ketika <i>use case</i> berakhir.
Extension	Informasi penggunaan <i>use case</i> lain yang dikembangkan dan bersifat opsional yang diikuti sertakan dalam sebuah <i>use case</i> .
Basic Flow	Alur normal yang dilalui ketika <i>use case</i> berjalan.
Alternative Flow	Alur opsional yang akan dilalui saat <i>use case</i> tidak berjalan sesuai dengan alur normal.
Subflow	Beberapa kelompok alur yang merupakan penyederhanaan alur normal <i>use case</i> yang terlalu terperinci.

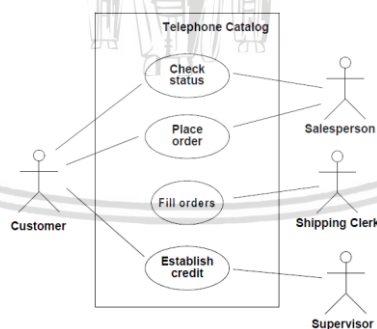
Sumber: Diadaptasi dari Bittner & Spence (2002)

2.10 Unified Modeling Language (UML)

UML atau *Unified Modeling Language* merupakan bahasa visual untuk menentukan, membangun dan mendokumentasikan artefak dari suatu sistem. Bahasa pemodelan ini bertujuan umum untuk dapat digunakan dengan semua objek utama dan *component method* dan dapat diterapkan pada semua domain aplikasi (Group, 2005). Diagram-diagram yang terdapat dalam UML yang dapat digunakan untuk menggambarkan sistem terdiri dari *use case diagram*, *behavior diagram*, *class diagram*, *implementation diagram*, *statechart diagram*, *activity diagram*, *interaction diagrams*, *collaboration diagram*, *component diagram*, *deployment diagram*, dll. Dalam penelitian ini memanfaatkan beberapa diagram diantaranya *use case diagram*, *class diagram* dan *sequence diagram*.

Use Case Diagram


2.10.1 *Use case* adalah sarana untuk menentukan kebutuhan atau persyaratan yang dibutuhkan oleh suatu sistem. Biasanya *use case* digunakan untuk menjelaskan persyaratan dari sebuah sistem, yaitu apa yang seharusnya dilakukan oleh sebuah sistem. Kunci dari sebuah *use case* adalah aktor, *use case* dan subjek. Subjeknya adalah sistem yang dipertimbangkan dan diberlakukannya *use case*. Pengguna dan sistem lainnya dapat berinteraksi dengan subjek yang direpresentasikan sebagai aktor. Aktor adalah entitas yang berada diluar sistem. Perilaku yang diminta dari subjek ditentukan oleh beberapa *use case* yang didefinisikan sesuai dengan kebutuhan aktor atau pengguna (Group, 2005). Gambar 2.3 memperlihatkan contoh diagram *use case*. Tabel 2.12 menjelaskan mengenai notasi-notasi diagram *use case*.

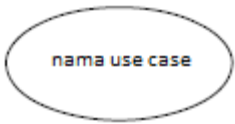
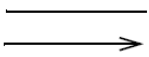
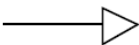
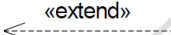
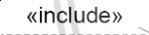


Gambar 2.3 Contoh Use Case Diagram

Sumber: Object Management Group (2005)

Tabel 2.12 Notasi pada Use Case Diagram

Notasi	Nama	Deskripsi
	Aktor	Aktor adalah peran yang diperankan oleh pengguna atau sistem lain yang berinteraksi

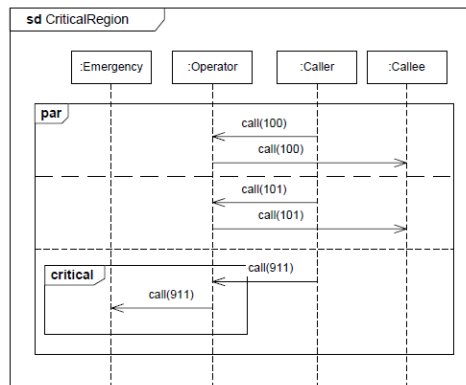
Notasi	Nama	Deskripsi
		dengan sistem. Aktor berada diluar dari sistem.
	<i>Use case</i>	<i>Use case</i> merupakan spesifikasi dari sekumpulan aksi yang dilakukan oleh sistem yang akan memberikan nilai untuk satu atau lebih aktor atau pemangku kepentingan sebuah sistem.
	Asosiasi	Notasi yang menggambarkan hubungan antara aktor dengan <i>use case</i> .
	Generalisasi	Notasi yang menggambarkan hubungan antar aktor. Menghubungkan aktor yang lebih spesifik dengan aktor yang umum.
	<i>Extends</i>	Notasi yang menggambarkan hubungan antar <i>use case</i> yang menunjukkan bahwa perilaku sebuah <i>use case</i> dapat dikembangkan (<i>extended</i>) oleh <i>use case</i> yang lain.
	<i>Include</i>	Notasi yang menggambarkan hubungan antar <i>use case</i> yang menunjukkan bahwa sebuah <i>use case</i> merupakan bagian dari <i>use case</i> lain.

2.10.2

Sumber: Diadaptasi dari Object Management Group (2005)

Sequence Diagram

Sequence diagram adalah diagram interaksi yang menggambarkan alur waktu pertukaran pesan antar objek. Sebuah *sequence diagram* menunjukkan satu set objek dan pesan yang dikirim dan diterima oleh objek-objek tersebut. *Sequence diagram* digunakan untuk menggambarkan tampilan dinamis dari suatu sistem. (Booch, et al., 1998). Gambar 2.4 merupakan contoh dari *sequence diagram*. Tabel 2.13 menjelaskan beberapa notasi dari *sequence diagram*.

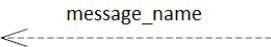
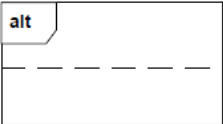
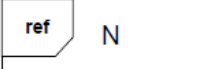


Gambar 2.4 Contoh Sequence Diagram

Sumber: Object Management Group (2005)

Tabel 2.13 Notasi pada Sequence Diagram

Notasi	Nama	Deskripsi
	<i>Frame</i>	Dalam notasi ini terdapat serangkaian objek atau <i>instances</i> yang saling bertukar pesan sebagai visualisasi aspek dinamis bagian dari sistem. Nama interaksi dituliskan di pojok kiri atas notasi <i>frame</i> .
	<i>Lifeline</i>	Notasi ini mewakili watau menggambarkan sebuah <i>instances</i> yang terlibat dalam sebuah interaksi pertukaran pesan.
	<i>Message (call)</i>	Notasi ini menggambarkan komunikasi yang bersifat sinkronis atau <i>synchronous</i> di antara <i>lifeline</i> . Notasi <i>call</i> secara khusus menggambarkan pemanggilan operasi atau metode. Apabila notasi ini digunakan untuk memodelkan pemanggilan operasi dengan argumen, maka argumen tersebut harus disertakan dalam penamaan notasi.
	<i>Message (asynchronous)</i>	Notasi menggambarkan komunikasi yang bersifat tidak sinkronis atau <i>asynchronous</i> di antara <i>lifeline</i> .

Notasi	Nama	Deskripsi
	<i>Message (reply)</i>	Notasi menggambarkan pesan balasan dari penggunaan notasi <i>synchronous message call</i> atau <i>asynchronous message call</i> .
	<i>Combined Fragment (Fragment)</i>	<p>Notasi ini digunakan untuk mengelompokkan interaksi kedalam fragmen. Setiap fragmen dapat memiliki beberapa <i>operand</i> yang masing-masing ditempatkan pada jalur yang dipisahkan oleh garis putus-putus di dalam fragmen.</p> <p>Varian <i>alternative</i> atau (alt) merupakan varian yang menggambarkan interaksi kondisional. Pada varian ini setidaknya satu <i>operand</i> akan dijalankan.</p> <p>Varian <i>option</i> atau (opt) merupakan varian yang menggambarkan interaksi kondisional. Varian ini hanya boleh memiliki 1 <i>operand</i>, yang bisa saja tidak dijalankan apabila kondisi pada <i>guard</i> tidak terpenuhi.</p> <p>Varian <i>loop</i> atau perulangan merupakan varian yang menggambarkan interaksi yang dilakukan berulang-ulang selama kondisi <i>operand</i> terpenuhi. Varian ini hanya boleh memiliki 1 <i>operand</i>.</p>
	<i>Interaction Use (Fragment)</i>	Notasi ini digunakan untuk merujuk sebuah interaksi. Pemodelan notasi ini menggunakan notasi <i>combined fragment</i> dengan nama <i>interaction operator</i> ref. Nama interaksi yang dirujuk diletakkan di tengah notasi.

Sumber: Diadaptasi dari Object Management Group (2005)

Class Diagram

2.10.3 Untuk menunjukkan keberadaan kelas dan hubungannya dalam suatu sistem digunakan diagram kelas. Sebuah diagram kelas tunggal mewakili atau merepresentasikan sebuah struktur kelas dari suatu sistem. Diagram kelas dapat digunakan untuk menunjukkan peran dan tanggung jawab entitas yang menggambarkan perilaku sistem saat analisis. Pada saat perancangan atau desain, diagram kelas digunakan untuk menangkap struktur-struktur kelas yang membentuk arsitektur sistem (Booch, et al., 2007). Penjelasan notasi *class diagram* terdapat pada Tabel 2.14. Notasi kelas memiliki struktur yang terdiri dari kolom nama kelas, kolom atribut dan kolom operasi.

Tabel 2.14 Notasi pada Class Diagram

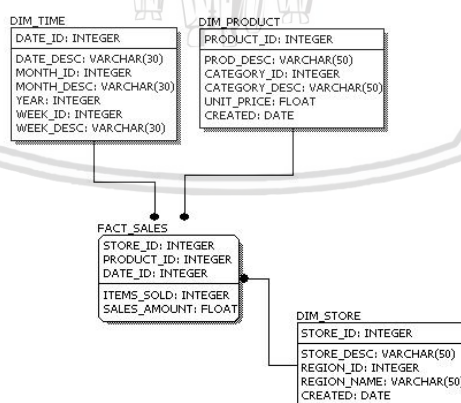
Notasi	Nama	Deskripsi
	<i>Class</i>	Notasi yang menggambarkan atau mewakili sebuah objek beserta atribut dan operasi yang dimiliki. Struktur notasi terdiri dari kolom nama kelas, kolom atribut, dan kolom operasi.
	<i>Association</i>	Notasi yang menggambarkan atau mewakili relasi sebuah objek dengan objek yang lain.
	<i>Generalization</i>	Notasi yang mewakili hubungan sebuah objek yang spesifik atau <i>subclass</i> dengan objek yang lebih umum atau <i>superclass</i> . <i>Subclass</i> mewarisi atribut dan operasi yang dimiliki oleh <i>superclass</i> .
	<i>Aggregation</i>	Notasi yang mewakili hubungan sebuah objek yang lebih luas dengan objek yang menjadi bagian dari objek tersebut. Objek yang lebih luas diletakkan pada sisi ujung notasi yang memiliki bentuk belah ketupak, sedangkan bagian dari objek diletakkan pada ujung yang lain.
	<i>Composition</i>	Notasi ini merupakan bentuk lain <i>aggregation</i> , akan tetapi dalam notasi ini bagian dari objek merupakan bagian

Notasi	Nama	Deskripsi
		hanya boleh 1 dari objek yang lebih besar.
+	<i>Public</i> (Visibility Kind)	Notasi yang menjelaskan bahwa atribut atau operasi bersifat <i>public</i> atau dapat digunakan secara langsung oleh semuanya.
#	<i>Protected</i> (Visibility Kind)	Notasi ini menjelaskan bahwa atribut atau operasi hanya dapat diakses oleh <i>instance</i> dari objek <i>subclass</i> .
-	<i>Private</i> (Visibility Kind)	Notasi ini menjelaskan bahwa atribut atau operasi hanya dapat diakses oleh objek yang memilikinya.

Sumber: Object Management Group (2005)

2.11 Physical Data Model

Pemodelan data adalah tindakan mengeksplorasi struktur struktur yang berorientasi data. Seperti model-model artefak pemodelan lainnya, pemodelan data dapat digunakan untuk berbagai tujuan, dari konseptual tingkat tinggi hingga model data fisik (PDM). *Physical Data Model* (PDM) secara konseptual mirip dengan pemodelan kelas. Tujuannya adalah untuk merancang skema internal dari basis data, menggambarkan tabel data, kolom data dari tabel tersebut dan hubungan antar tabel (Inc., 2013).



Gambar 2.5 Contoh *Physical Data Model*

Sumber: 1keydata (2009)

Pemodelan data pada PDM dapat dilakukan mengacu pada hasil pemodelan kelas yang digambarkan dalam diagram kelas. Contoh PDM terdapat dalam Gambar 2.5. Menurut Booch, et al. (1998), pemodelan PDM atau *physical*

database dapat dilakukan dengan 3 strategi. Tiga strategi dibawah ini dapat digunakan salah satu atau mengkombinasikan antara ketiga strategi dibawah ini. Berikut adalah tiga strategi tersebut:

1. Menentukan tabel terpisah untuk setiap kelas. Ini hal sederhana akan tetapi sulit dilakukan saat menambah kelas turunan baru atau merubah kelas induk.
2. Memperkecil kemungkinan *inheritance* sehingga semua *instance* dari kelas apapun memiliki keadaan yang sama. Kelemahan dari strategi ini adalah penyimpanan menjadi berlebihan.
3. Memisahkan status *parent class* dan *subclass* ke dalam tabel yang berbeda. Kelemahan dari strategi ini adalah akan membutuhkan banyak *crosstable joins*.

2.12 Model-View-Controller

MVC atau *Model-View-Controller* adalah memisahkan interaksi dari data sistem, sistem terstruktur menjadi tiga komponen yang berinteraksi satu sama lain. Komponen *model* mengelola data sistem dan operasi yang terkait pada data tersebut. Komponen tampilan atau *view* menentukan dan mengatur bagaimana data disajikan kepada pengguna. Komponen *controller* mengelola interaksi pengguna (misalnya penekanan tombol, klik *mouse* dan lain sebagainya) dan meneruskan interaksi ke tampilan dan *model* (Sommerville, 2011).

2.13 Pengujian Perangkat Lunak

Pengujian dilakukan untuk menunjukkan bahwa suatu program atau perangkat lunak melakukan apa yang telah direncanakan atau memenuhi tujuan dan untuk menemukan cacat program sebelum mulai digunakan (Sommerville, 2011). Sommerville dalam bukunya juga menambahkan pengujian perangkat lunak mempunyai 2 tujuan:

1. Untuk menunjukkan kepada pengembang dan pelanggan atau pengguna bahwa perangkat lunak telah memenuhi persyaratan atau kebutuhan.
2. Untuk menemukan situasi saat perilaku perangkat lunak salah, tidak diinginkan atau tidak sesuai dengan spesifikasi.

Pengujian Validasi

Validasi dapat didefinisikan dalam berbagai macam cara, akan tetapi definisi yang sederhana yaitu validasi berhasil ketika fungsi perangkat lunak sesuai dengan harapan pengguna atau pelanggan (Pressman, 2010). Pengujian validasi dilakukan untuk memperagakan perangkat lunak atau sistem telah memenuhi persyaratan. Pressman menjelaskan bahwa setelah pengujian dilakukan atau dilaksanakan akan didapatkan dua hasil:

1. Fungsi sesuai dengan spesifikasi dan diterima.

2. Penyimpangan dari spesifikasi terungkap dan kekurangan akan dicatat.

Penyimpangan atau kesalahan yang ditemukan pada tahap ini jarang dikoreksi atau diperbaiki sebelum jadwal perilisan. Untuk membantu mengatasi penyimpangan saat pengujian validasi, penguji harus mengacu pada *use case* yang menjadi bagian dari model kebutuhan (Pressman, 2010).

Pengujian Responsif

Pengujian desain yang responsif berarti menguji situs web atau URL dari perangkat yang berbeda-beda. Secara praktis, tidak mungkin menguji situs web

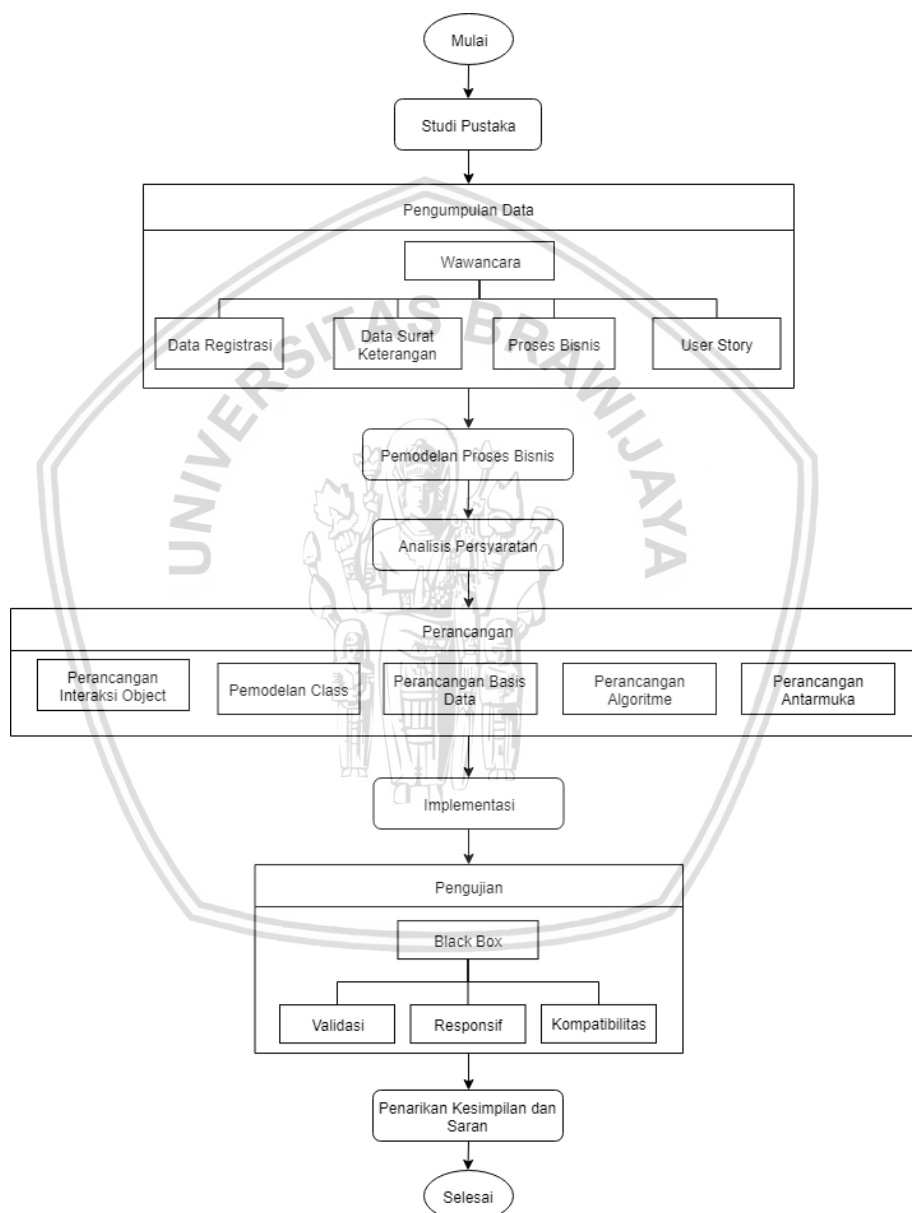
- 2.13.2 yang responsif sepenuhnya dalam satu perangkat, karena itu perlu menyiapkan berbagai perangkat dengan berbagai ukuran layar. Cara yang mungkin untuk pengujian responsif adalah dengan mengubah ukuran jendela peramban sesuai dengan skenario pengujian (Laxmi, 2018).

Pengujian Kompatibilitas

- 2.13.3 Pengujian kompatibilitas dilakukan terhadap antarmuka pengguna pada aplikasi berbasis web. Pengujian kompatibilitas dilakukan untuk mengetahui aplikasi memiliki masalah ketika berjalan pada komputer, perangkat tampilan, sistem operasi, aplikasi peramban, atau kecepatan jaringan yang berbeda atau tidak (Pressman, 2010). Pengujian kompatibilitas dilakukan dengan mendefinisikan atau menjelaskan sekumpulan konfigurasi pada perangkat pengujian dan menguji aplikasi atau sistem pada perangkat tersebut. Konfigurasi yang dimaksud dapat meliputi peramban yang berbeda, sistem operasi yang berbeda, perangkat tampilan yang berbeda maupun kecepatan jaringan yang berbeda.

BAB 3 METODOLOGI

Pada bab atau bagian ini dijelaskan mengenai langkah atau tahap yang akan dilakukan. Tahapan tersebut adalah studi pustaka, pengumpulan data dari pemangku kepentingan, pemodelan proses bisnis *as-is* dan *to-be*, analisis persyaratan, perancangan, implementasi, pengujian serta kesimpulan dan saran. Pada Gambar 3.1 dijelaskan diagram alir runtutan pengerjaan penelitian ini.



Gambar 3.1 Diagram Alur Metode Penelitian

3.1 Studi Pustaka

Pada tahap ini dilakukan pembelajaran mengenai beberapa literatur terhadap segala hal yang berhubungan dengan penelitian. Pembelajaran mengenai

beberapa literatur digunakan guna mempelajari berbagai referensi yang diambil dari berbagai sumber seperti : jurnal, buku, hasil konferensi, laporan penelitian sebelumnya yang sejenis, artikel, dll. Tahap ini bertujuan untuk mendapatkan pemahaman yang berguna dalam penelitian tentang pengembangan Sistem Informasi Pelayanan Surat Keterangan.

3.2 Pengumpulan Data

Pengumpulan data dilakukan guna mengumpulkan data dan informasi yang dapat membantu penelitian. Metode wawancara digunakan untuk pengumpulan data. Wawancara dilakukan dengan narasumber beberapa pegawai Pemerintah Desa Legundi. Wawancara dilakukan secara bebas dan terbuka dimana responden diberikan kesempatan dan kebebasan secara penuh untuk memberikan jawaban tanpa ada batasan sehingga peneliti dapat menggali informasi sebanyak-banyaknya. Informasi yang digali berkaitan dengan topik penelitian yang dilakukan untuk mendapatkan atau memperoleh kebutuhan sistem yang sesuai sehingga dapat memberikan solusi terhadap permasalahan yang ada.

Berikut adalah data hasil dari wawancara dengan pegawai Pemerintah Desa Legundi.

1. Data Registrasi

Data Registrasi adalah data yang berisi identitas penduduk yang melakukan permohonan surat keterangan beserta nomor surat.

2. Data Surat Keterangan

Data surat keterangan adalah template dokumen digital surat keterangan.

3. Proses Bisnis

Proses bisnis atau tahapan dalam permohonan surat keterangan saat ini sebelum adanya sistem.

4. User Story

User story berkaitan dengan apa yang diinginkan stakeholder kenapa diperlukan. User story dapat membantu untuk pembuatan persyaratan sistem.

3.3 Pemodelan Proses Bisnis

Business Process Model and Notation (BPMN) digunakan untuk menggambarkan proses yang ada pada saat ini serta rekomendasi perbaikan proses bisnis. Tujuan dari pemodelan proses bisnis adalah memudahkan peneliti, pemangku kepentingan, dan pengguna yang terlibat guna mengetahui proses

bisnis yang sedang berjalan serta mengidentifikasi proses yang bermasalah. Pemodelan proses bisnis menghasilkan model proses bisnis *As-Is* dan *To-Be*.

3.4 Analisis Persyaratan

Berdasarkan hasil pengumpulan data yang telah dilakukan selanjutnya dilakukan analisis persyaratan. Analisis persyaratan terdiri dari identifikasi tipe pemangku kepentingan, analisis masalah yang ada, identifikasi kebutuhan pemangku kepentingan dan pengguna, identifikasi fitur yang akan terdapat pada sistem, persyaratan fungsional sistem, persyaratan non fungsional sistem dan pemodelan *use case*. Analisis persyaratan dilakukan untuk mengetahui apa yang dibutuhkan dan diinginkan pengguna terhadap sistem yang dikembangkan.

3.5 Perancangan

Pada tahap atau bagian ini dilakukan perancangan sistem dengan pemodelan aliran data berdasarkan tahap analisis kebutuhan. Perancangan sistem informasi dilakukan untuk memodelkan sistem berdasarkan hasil analisis persyaratan ke dalam notasi-notasi pada *Unified Modeling Language* (UML). Perancangan sistem menghasilkan pemodelan interaksi objek pada *sequence diagram*, pemodelan objek pada *class diagram*, perancangan basis data pada PDM, perancangan algoritme ke dalam *pseudocode* serta perancangan antarmuka pengguna.

3.6 Implementasi

Implementasi sistem informasi adalah tahap pembuatan Sistem Informasi Pelayanan Surat Keterangan yang mengacu pada hasil perancangan sistem. Pengembangan sistem menggunakan bahasa pemrograman PHP dan Javascript, dengan kerangka kerja web Codeigniter serta basis data menggunakan mySql. Hasil dari pembuatan sistem informasi adalah suatu Sistem Informasi Pelayanan Surat Keterangan berbasis web yang menyediakan fitur sesuai dengan yang ada pada tahap analisis kebutuhan. Pada tahap implementasi juga dilakukan pendokumentasian. Informasi mengenai perangkat lunak dan perangkat keras yang digunakan, hasil implementasi *pseudocode* atau algoritme serta hasil implementasi antarmuka pengguna didokumentasikan pada tahap implementasi.

3.7 Pengujian

Pengujian dilakukan dengan melakukan uji coba yang dilakukan oleh pengguna bertujuan untuk menguji tingkat kesuksesan sebuah perangkat lunak. Menguji sistem dengan mencoba sistem yang telah jadi dan dinilai ketika digunakan. Pengujian yang dilakukan menggunakan pengujian yaitu black box yaitu dengan metode pengujian validasi, responsif dan kompatibilitas.

3.8 Penarikan Kesimpulan dan Saran

Penarikan kesimpulan dan saran dilakukan setelah semua tahapan selesai dilakukan. Apabila semua tahap sudah diselesaikan maka perlu adanya evaluasi atau pengecekan ulang pada tiap metode pengujian yang digunakan agar mendapatkan data yang valid. Kesimpulan dari pengembangan sistem yang telah dilakukan didapatkan dengan cara analisis pengujian sistem. Setelah kesimpulan didapatkan maka tahap akhir adalah saran yang berguna untuk memperbaiki celah dan kesalahan yang terjadi dari penelitian yang telah dilakukan sehingga akan menyempurnakan penulisan dan dapat digunakan sebagai landasan dalam pengembangan serta memperbaiki kelemahan sistem selanjutnya.



BAB 4 PEMODELAN PROSES BISNIS DAN ANALISIS PERSYARATAN SISTEM

Bab ini menjelaskan tentang pemodelan proses bisnis dan analisa persyaratan dari Sistem Informasi Pelayanan Surat Keterangan Pemerintah Desa Legundi yang dilakukan.

4.1 Pemodelan Proses Bisnis

Pemodelan proses bisnis menjelaskan mengenai proses bisnis yang berjalan pada instansi terkait. Terdapat proses bisnis sebelum diterapkannya Sistem Informasi Pelayanan Surat Keterangan dan proses bisnis setelah diterapkannya Sistem Informasi Pelayanan Surat Keterangan. Pemodelan proses bisnis menghasilkan model proses bisnis *As-Is* dan *To-Be* pada setiap kegiatan dalam instansi terkait.

4.1.1 Proses Bisnis *As-Is*

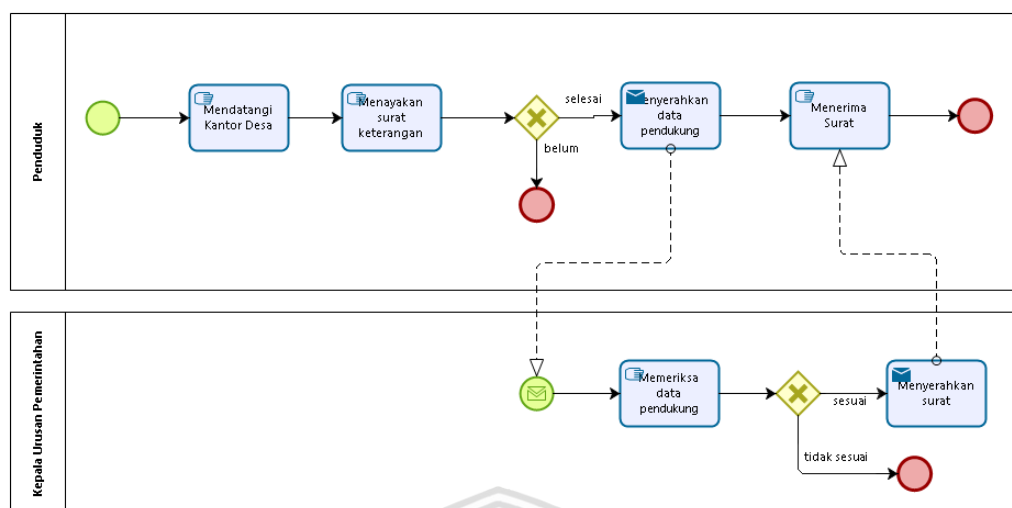
Berdasarkan hasil wawancara dengan sekretaris Desa Legundi, didapatkan alur kegiatan pembuatan surat keterangan berjalan pada Pemerintah Desa Legundi.

4.1.1.1 Proses Bisnis *As-Is* Memohon Surat Keterangan

Proses bisnis memohon surat keterangan dimulai dengan penduduk mendatangi kantor desa, selanjutnya penduduk memohon surat keterangan. Selanjutnya menyerahkan data surat termasuk KTP sebagai syarat pembuatan surat keterangan. Setelah itu kepala urusan pemerintahan memverifikasi data. Jika data diterima maka kepala urusan pemerintahan membuat nomor surat dengan melihat pada buku registrasi kemudian menulis pada buku registrasi. Setelah itu membuat draft surat dan mencetak surat. Setelah mencetak surat meminta paraf persetujuan pada sekretaris desa. Jika sekretaris desa menyetujui maka selanjutnya kepala urusan pemerintahan meminta pengesahan surat kepada kepala desa. Berikut ini pada Penggambaran notasi BPMN untuk menggambarkan bagaimana proses bisnis untuk permohonan surat keterangan digambarkan pada Lampiran A.1.

4.1.1.2 Proses Bisnis *As-Is* Periksa Perkembangan Permohonan Surat Keterangan

Penggambaran notasi BPMN untuk menggambarkan proses bisnis periksa hasil permohonan surat keterangan terdapat pada Gambar 4.1.



Gambar 4.1 Proses Bisnis As-Is Periksa Perkembangan Permohonan Surat Keterangan

Proses bisnis dimulai dengan penduduk mendatangi kantor desa kemudian menanyakan surat keterangan. Apabila surat belum selesai maka selesai atau penduduk harus menunggu kembali. Apabila selesai maka penduduk menyerahkan data pendukung yaitu KTP. Setelah itu, kepala urusan pemerintahan memeriksa data pendukung, apabila sesuai maka menyerahkan surat dan penduduk menerima surat.

4.1.2 Proses Bisnis To-Be

Berdasarkan hasil wawancara yang dilakukan oleh penulis, didapatkan *user story* yang akan mengubah proses bisnis As-Is. *User story* yang didapatkan adalah sebagai berikut.

1. Sebagai kepala urusan pemerintahan saya membutuhkan sistem yang bisa menyimpan data surat keterangan sehingga saya tidak perlu menyimpan data surat keterangan dalam buku registrasi.
2. Sebagai kepala urusan pemerintahan saya membutuhkan sistem yang bisa mencari dengan mudah data yang telah tersimpan sehingga tidak perlu memerlukan waktu yang lama untuk mencari data.
3. Sebagai kepala urusan pemerintahan saya membutuhkan sistem yang bisa membuat nomor surat secara otomatis sehingga tidak perlu memeriksa jenis bidang serta nomor surat sebelumnya.
4. Sebagai kepala urusan pemerintahan saya membutuhkan sistem yang langsung bisa mencetak surat tanpa membuat draft surat terlebih dahulu sehingga tidak melakukan dua kali pekerjaan.

5. Sebagai penduduk saya membutuhkan sistem yang bisa memohon surat dimana saja dan kapan saja sehingga tidak perlu datang ke kantor desa.
6. Sebagai penduduk saya membutuhkan sistem yang bisa memeriksa hasil permohonan surat keterangan dimana saja dan kapan saja sehingga saya tidak perlu datang ke kantor desa berkali-kali ketika surat keterangan belum selesai.

Berdasarkan hasil analisis permasalahan dan *user story*, diperoleh solusi terhadap proses bisnis yang dibuat dengan diagram BPMN.

4.1.2.1 Proses Bisnis *To-Be* Permohonan Surat Keterangan

Berdasarkan hasil analisa permasalahan yang dijelaskan sebelumnya dan berdasarkan *user story* yang didapatkan, proses bisnis untuk memohon surat keterangan akan diubah. Model proses bisnis yang baru ditunjukkan pada Lampiran A.2. Tabel 4.1 menjelaskan perubahan aktivitas.

Tabel 4.1 Perubahan Aktivitas Proses Bisnis Permohonan Surat Keterangan

Unit Bisnis	Proses Bisnis <i>As-Is</i>	Proses Bisnis <i>To-Be</i>	Keterangan
Kepala Urusan Pemerintahan	Membuat nomor surat	-	Dieliminasi
	Menulis data dalam buku registrasi	-	Dieliminasi
	Membuat draft surat keterangan	-	Dieliminasi
Penduduk	Mendatangi kantor desa	Membuka website Pelayanan Surat Keterangan	Mendatangi kantor desa bisa diganti dengan membuka website pelayanan surat keterangan

Pada Lampiran A.2 dijelaskan proses bisnis baru yang digambarkan dengan diagram BPMN. Proses bisnis dimulai dengan penduduk memilih metode apa yang akan dilakukan saat ingin memohon surat keterangan. Apabila penduduk ingin datang ke kantor desa maka penduduk mendatangi kantor desa dan menyerahkan data surat keterangan beserta KTP sebagai syarat permohonan surat keterangan. Setelah itu, kepala urusan pemerintahan mengisi data pada formulir yang terdapat pada sistem dan menyimpan data. Apabila penduduk tidak datang ke kantor desa maka membuka website dan mengisi formulir surat keterangan dan menyimpan

data. Apabila data telah masuk ke sistem maka kepala urusan pemerintahan akan memeriksa atau memverifikasi permohonan surat yang masuk pada sistem. Apabila ditolak maka mengubah status surat menjadi ditolak. Apabila diterima dan ada kesalahan maka kepala urusan pemerintahan akan mengubah data yang salah. Apabila diterima dan tidak ada kesalahan maka kepala urusan pemerintahan mengubah status surat menjadi terverifikasi. Setelah itu sekretaris desa akan memeriksa surat yang sudah terverifikasi. Jika surat disetujui maka sekretaris desa mengganti status disetujui. Jika ditolak maka sekretaris desa mengubah status ditolak. Setelah itu kepala urusan pemerintahan akan mencetak surat dan meminta tanda tangan ke kepala desa. Setelah itu kepala desa mengesahkan surat dan kepala urusan pemerintahan mengubah status selesai. Dalam analisis waktu yang dilakukan pada hasil simulasi bpmn didapatkan proses bisnis *To-Be* lebih cepat dari proses bisnis *As-Is*. Hasil dari simulasi waktu pada *software* bizagi modeler dari 100 kejadian pada proses *As-Is* total waktu pada *pool* penduduk adalah 23 jam 20 menit dan pada *pool* kantor desa adalah 22 jam 24 menit. Sedangkan pada proses *To-Be* total waktu pada *pool* penduduk adalah 12 jam 19 menit dan pada *pool* kantor desa adalah 11 jam 13 menit.

4.1.2.2 Proses Bisnis *To-Be* Periksa Perkembangan Permohonan Surat

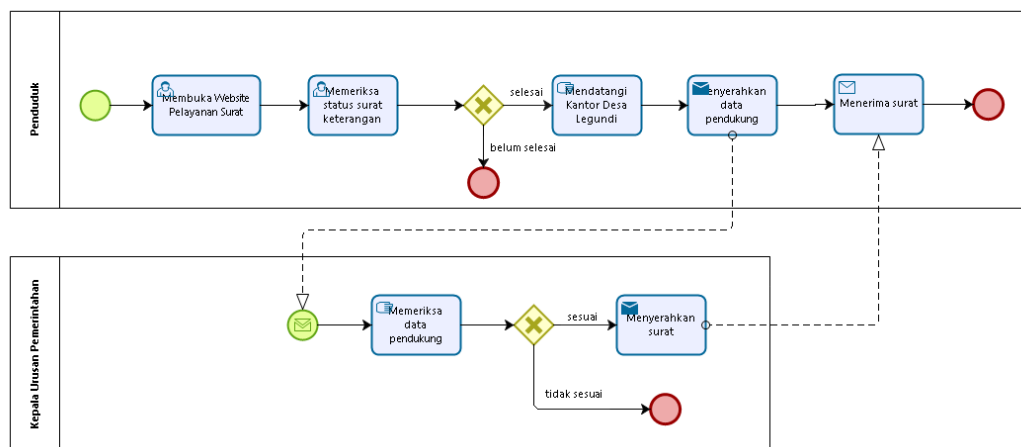
Keterangan

Berdasarkan hasil analisa permasalahan yang sudah dijelaskan sebelumnya, proses bisnis untuk periksa hasil permohonan surat keterangan akan diubah. Dalam analisis waktu yang dilakukan pada hasil simulasi BPMN didapatkan proses bisnis *To-Be* lebih cepat dari proses bisnis *As-Is*. Hasil dari simulasi waktu pada *software* bizagi modeler dari 100 kejadian pada proses *As-Is* total waktu pada *pool* penduduk adalah 1 hari 4 jam 15 menit dan pada *pool* kepala urusan pemerintahan adalah 4 jam 39 menit. Sedangkan pada proses *To-Be* total waktu pada *pool* penduduk adalah 1 hari 1 jam 48 menit dan pada *pool* kepala urusan pemerintahan adalah 3 jam 32 menit. Model proses bisnis yang baru ditunjukkan pada Gambar 4.2.

Tabel 4.2 Perubahan Aktivitas Proses Bisnis Periksa Hasil Permohonan Surat

Keterangan

Unit Bisnis	Proses Bisnis <i>As-Is</i>	Proses Bisnis <i>To-Be</i>	Keterangan
Penduduk	Mendatangi kantor desa	Membuka website Pelayanan Surat Keterangan	Mendatangi kantor desa diganti menjadi membuka website Pelayanan Surat Keterangan



Gambar 4.2 Proses Bisnis To-Be Periksa Hasil Permohonan Surat Keterangan

Model proses bisnis baru digambarkan dengan diagram BPMN pada Gambar 4.2. Proses bisnis dimulai dengan penduduk membuka website sistem pelayanan surat keterangan. Setelah itu penduduk memeriksa status surat keterangan dengan nomor registrasi surat. Jika belum selesai maka penduduk menunggu surat keterangan. Jika sudah selesai maka penduduk mendatangi kantor desa dan menyerahkan data pendukung yaitu KTP. Kemudian kepala urusan pemerintahan memeriksa data pendukung. Jika sesuai maka kepala urusan pemerintahan menyerahkan surat dan penduduk menerima surat keterangan.

4.2 Analisis Persyaratan

Pada tahap ini dijelaskan analisis persyaratan yang dibutuhkan oleh sistem.

4.2.1 Identifikasi Tipe Pemangku Kepentingan

Untuk mengetahui dan mengelompokkan pemangku kepentingan sesuai dengan karakteristik dan hubungan pemangku kepentingan dengan sistem dilakukan identifikasi pemangku kepentingan. Informasi dan data mengenai pemangku kepentingan diperoleh dari hasil wawancara dengan Pegawai Pemerintah Desa Legundi dan masyarakat desa Legundi. Identifikasi tipe pemangku kepentingan digunakan sebagai informasi untuk menganalisis masalah dari berbagai pemangku kepentingan yang berbeda pada lokasi studi kasus. Penjelasan tipe pemangku kepentingan untuk penelitian ini terdapat pada Tabel 4.3.

Tabel 4.3 Tipe Pemangku Kepentingan

Tipe Pemangku Kepentingan	Deskripsi	Pemangku Kepentingan
Pengguna	Seseorang yang menggunakan sistem informasi secara langsung dan berperan sebagai aktor pada <i>use case</i> .	<ul style="list-style-type: none"> • Sekretaris Desa • Kepala Urusan Pemerintahan • Penduduk Desa Legundi
Pengembang	Organisasi atau seseorang yang melakukan pengembangan sistem informasi pelayanan surat keterangan.	Peneliti
Pihak yang berwenang	Organisasi atau seseorang yang mampu mendukung pengembangan sistem informasi pelayanan surat keterangan dengan cara memberikan informasi mengenai regulasi yang berjalan, sehingga solusi yang dikembangkan dapat berjalan sesuai dengan aturan yang berlaku.	Pemerintah Desa Legundi
Pelanggan	Organisasi atau seseorang yang mendapatkan manfaat dari sistem informasi pelayanan surat keterangan.	<ul style="list-style-type: none"> • Pemerintah Desa Legundi • Peduduk Desa Legundi

4.2.2 Analisis Masalah

Untuk memahami masalah yang terjadi dan masalah yang harus diselesaikan oleh pemangku kepentingan analisis masalah dilakukan. Pada penelitian ini, analisis masalah dilakukan dengan analisis hasil wawancara dengan sekretaris Desa Legundi dan penduduk Desa Legundi. Pernyataan masalah yang didapatkan dari analisis hasil wawancara pegawai Pemerintah Desa Legundi terdapat pada Tabel 4.4.

Tabel 4.4 Pernyataan Masalah 1

Masalah	<ul style="list-style-type: none"> • Pencacatan data surat keterangan masih dalam buku registrasi. • Terdapat dua kali kerja yaitu pencatatan pada buku registrasi dan pembuatan draft dokumen digital. • Pembuatan nomor masih menggunakan cara melihat nomor sebelumnya.
Mempengaruhi	Pengguna, pihak yang berwenang dan pelanggan.
Berdampak pada	<ul style="list-style-type: none"> • Terdapat data yang hilang yaitu data yang berada pada template dokumen digital. • Kesulitan pencarian data surat keterangan.. • Kesalahan pada pembuatan nomor surat. • Kesalahan pegawai lupa tidak menulis pada buku registrasi. • Dokumen buku registrasi bisa hilang.
Solusi sukses	Menyediakan layanan untuk pencatatan sekaligus pembuatan surat keterangan

Pernyataan masalah selanjutnya didapatkan dengan wawancara terhadap beberapa penduduk Desa Legundi. Berikut pernyataan masalah kedua yang akan dijelaskan pada Tabel 4.5.

Tabel 4.5 Pernyataan Masalah 2

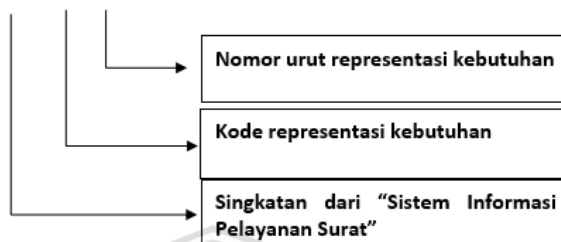
Masalah	<ul style="list-style-type: none"> • Permohonan surat keterangan masih harus datang ke Kantor Desa Legundi. • Pengecekan hasil permohonan surat keterangan masih harus datang ke Kantor Desa Legundi.
Mempengaruhi	Pengguna dan pelanggan
Berdampak pada	<ul style="list-style-type: none"> • Waktu yang diperlukan menjadi lebih banyak.
Solusi sukses	Menyediakan layanan permohonan surat keterangan dan pengecekan hasil permohonan surat keterangan yang bisa diakses dimana saja dan kapan saja

4.2.3 Identifikasi Kebutuhan Pemangku Kepentingan dan Pengguna

Kebutuhan pemangku kepentingan dan pengguna didapatkan melalui analisis pernyataan masalah dan hasil wawancara dengan penduduk Desa Legundi dan

Sekretaris Desa Legundi. Sebagai dasar informasi tahap analisis fitur yang merupakan bagian untuk mengidentifikasi atau mengenali solusi yang ditawarkan guna memenuhi kebutuhan pengguna digunakan hasil identifikasi kebutuhan. Dibawah ini adalah keterangan kodifikasi dari masing-masing kebutuhan pengguna.

Kode : SIPS-KB -xx



Pada Tabel 4.6 akan diinformasikan mengenai kode identitas, kebutuhan pengguna, pemangku kepentingan, situasi saat ini serta solusi yang ditawarkan untuk memenuhi kebutuhan yang telah teridentifikasi.

Tabel 4.6 Identifikasi Kebutuhan Pengguna

Kode Kebutuhan	Kebutuhan Pengguna	Pemangku Kepentingan	Situasi Saat Ini	Solusi Yang Ditawarkan
SIPS-KB-01	Sistem harus menyediakan pencatatan dan pengelolaan data surat keterangan	Kepala Urusan Pemerintahan	Pencatatan data surat keterangan masih berada dalam buku registrasi	Sistem Informasi yang menyediakan pencatatan data surat keterangan secara digital di dalam sebuah basis data
SIPS-KB-02	Sistem harus menyediakan pembuatan nomor surat secara otomatis saat surat keterangan dibuat	Kepala Urusan Pemerintahan	Pembuatan nomor surat masih secara manual	Sistem Informasi yang menyediakan pembuatan nomor surat secara otomatis
SIPS-KB-03	Sistem harus mempermudah pencarian data	Kepala Urusan Pemerintahan,	Pencarian data masih melihat pada	Sistem Informasi yang mempermudah

Kode Kebutuhan	Kebutuhan Pengguna	Pemangku Kepentingan	Situasi Saat Ini	Solusi Yang Ditawarkan
	surat keterangan	Sekretaris Desa	buku registrasi	pencarian data surat keterangan dengan kata kunci tertentu
SIPS-KB-04	Sistem harus menyediakan layanan permohonan surat dimana saja dan kapan saja	Penduduk Desa Legundi	Permohonan surat keterangan masih harus datang ke Kantor Desa Legundi	Sistem Informasi menyediakan layanan permohonan surat yang dapat diakses dimana saja dan kapan saja
SIPS-KB-05	Sistem harus menyediakan layanan pengecekan hasil permohonan surat keterangan	Penduduk Desa Legundi	Pengecekan hasil permohonan surat masih harus datang ke Kantor Desa Legundi	Sistem Informasi menyediakan layanan pengecekan hasil permohonan surat keterangan yang dapat diakses dimana saja dan kapan saja

4.2.4 Identifikasi Pengguna

Untuk mengetahui pengguna yang akan menggunakan sistem secara langsung dilakukan identifikasi pengguna. Pengguna sistem adalah bagian dari kelompok kategori pengguna dalam identifikasi tipe pemangku kepentingan. Pada Tabel 4.7 dijelaskan mengenai identifikasi pengguna. Informasi mengenai identifikasi pengguna digunakan untuk mengidentifikasi aktor yang akan dimodelkan ke dalam diagram *use case*.

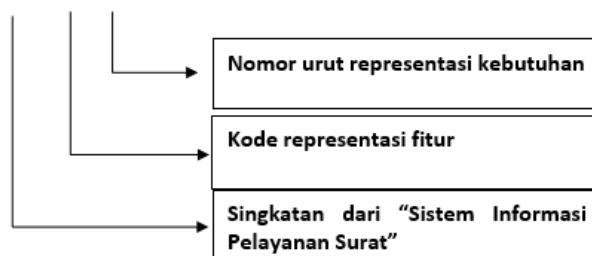
Tabel 4.7 Identifikasi Pengguna

Tipe Pemangku Kepentingan	Tipe Pengguna	Deskripsi
Pengguna	Sekretaris Desa	Pegawai Pemerintah Desa Legundi yang menjadi penanggung jawab kegiatan dalam pemerintahan desa.
	Kepala Urusan Pemerintahan	Pegawai Pemerintah Desa Legundi yang menjadi penanggung jawab dalam kegiatan permohonan surat keterangan.
	Penduduk Desa Legundi	Penduduk yang bertempat tinggal di Desa Legundi sesuai KTP yang berlaku.

4.2.5 Identifikasi Fitur

Identifikasi fitur mewakili solusi yang ditawarkan guna memenuhi kebutuhan pengguna. Identifikasi fitur diharapkan dapat menyelesaikan masalah yang dihadapi oleh pemangku kepentingan. Hasil dari identifikasi fitur adalah deskripsi singkat mengenai layanan yang diberikan atau disediakan oleh sistem guna memenuhi kebutuhan pengguna. Dibawah ini merupakan keterangan kodifikasi fitur.

Kode : SIPS-FT-xx



Identifikasi persyaratan fungsional dan non-fungsional dilakukan dengan mengacu pada hasil identifikasi fitur. Tabel 4.8 menjelaskan mengenai kode fitur dan deskripsi fitur.

Tabel 4.8 Hasil Identifikasi Fitur

Kode Fitur	Deskripsi Fitur	Level
SIPS-FT-01	Sistem harus dapat memastikan bahwa data yang digunakan dalam sistem harus terlindung dari akses yang tidak berwenang.	<i>Should</i>
SIPS-FT-02	Sistem dapat menambahkan atau melakukan pencatatan data surat keterangan.	<i>Must</i>
SIPS-FT-03	Sistem dapat mengganti status surat keterangan menjadi terverifikasi atau ditolak.	<i>Must</i>
SIPS-FT-04	Sistem dapat mengubah data surat keterangan	<i>Must</i>
SIPS-FT-05	Sistem dapat digunakan untuk mengganti status surat keterangan menjadi disetujui atau ditolak	<i>Must</i>
SIPS-FT-06	Sistem dapat digunakan untuk mengganti status menjadi selesai	<i>Must</i>
SIPS-FT-07	Sistem dapat menghapus data surat keterangan	<i>Must</i>
SIPS-FT-08	Sistem dapat mencetak surat keterangan	<i>Must</i>
SIPS-FT-09	Sistem dapat membuat nomor surat secara otomatis	<i>Must</i>
SIPS-FT-10	Sistem dapat digunakan untuk mencari data surat keterangan berdasarkan kata kunci tertentu.	<i>Must</i>
SIPS-FT-11	Sistem dapat digunakan untuk mengganti atau mengubah data pengguna yang menggunakan sistem.	<i>Should</i>
SIPS-FT-12	Sistem dapat digunakan untuk memeriksa perkembangan surat keterangan.	<i>Must</i>
SIPS-FT-13	Sistem dapat digunakan untuk membuat arsip file berformat <i>spreadsheet</i> data registrasi surat dengan status disetujui.	<i>Should</i>
SIPS-FT-14	Sistem harus dapat diakses dimana saja dan kapan saja	<i>Must</i>

Fitur-fitur yang terdapat pada penelitian ini dikelompokkan berdasarkan keterkaitan dengan kebutuhan pengguna. Pengelompokan ini menggambarkan bahwa sebuah fitur adalah solusi guna memenuhi kebutuhan tertentu. Pada Tabel 4.9 terdapat pengeleompokan fitur dengan kebutuhan pengguna.

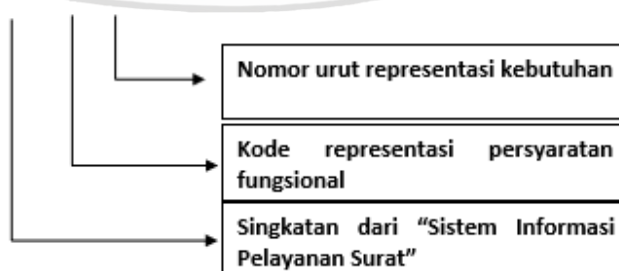
Tabel 4.9 Hubungan Fitur dengan Kebutuhan Pengguna

Fitur	Kebutuhan Pengguna
SIPS-FT-01	SIPS-KB-01
SIPS-FT-02	SIPS-KB-01, SIPS-KB-4
SIPS-FT-03	SIPS-KB-01
SIPS-FT-04	SIPS-KB-01
SIPS-FT-05	SIPS-KB-01
SIPS-FT-06	SIPS-KB-01
SIPS-FT-07	SIPS-KB-01
SIPS-FT-08	SIPS-KB-01
SIPS-FT-09	SIPS-KB-02
SIPS-FT-10	SIPS-KB-03
SIPS-FT-11	SIPS-KB-01
SIPS-FT-12	SIPS-KB-05
SIPS-FT-13	SIPS-KB-01
SIPS-FT-14	SIPS-KB-04, SIPS-KB-05

4.2.6 Persyaratan Fungsional

Persyaratan fungsional adalah kemampuan yang harus dipenuhi oleh sistem agar fitur sistem dapat berjalan dan dapat memenuhi kebutuhan pengguna. Hubungan antara fitur dan persyaratan fungsional ditunjukkan pada Tabel 4.10. Dibawah ini adalah kodifikasi persyaratan fungsional.

Kode : SIPS-PF-xx



Tabel 4.10 Persyaratan Fungsional

Kode Fitur	Kode Persyaratan Fungsional	Deskripsi
SIPS-FT-01	SIPS-PF-01	Sistem dapat menampilkan antarmuka otentifikasi pengguna.
	SIPS-PF-02	Sistem dapat mengotentifikasi pengguna berdasarkan data pengguna yang tersimpan.
	SIPS-PF-03	Sistem dapat menyimpan <i>session</i> berdasarkan data pengguna yang sedang aktif.
	SIPS-PF-04	Sistem dapat menghapus data <i>session</i> pengguna yang sedang aktif
SIPS-FT-02	SIPS-PF-05	Sistem dapat menampilkan formulir data surat keterangan
	SIPS-PF-06	Sistem dapat menambah atau menyimpan data surat keterangan ke dalam basis data dengan status diterima
	SIPS-PF-07	Sistem dapat menampilkan nomor registrasi surat keterangan setelah data surat keterangan tersimpan
SIPS-FT-03	SIPS-PF-08	Sistem dapat menampilkan daftar surat keterangan dengan status diterima
	SIPS-PF-09	Sistem dapat menampilkan detail surat keterangan dengan status diterima
	SIPS-PF-10	Sistem dapat mengganti status surat dari diterima menjadi terverifikasi
	SIPS-PF-11	Sistem dapat mengganti status surat dari diterima menjadi ditolak
SIPS-FT-04	SIPS-PF-12	Sistem dapat menampilkan formulir surat keterangan yang telah terisi data surat keterangan
	SIPS-PF-13	Sistem dapat mengubah data surat keterangan yang telah diperbarui
SIPS-FT-05	SIPS-PF-14	Sistem dapat menampilkan daftar surat keterangan dengan status terverifikasi
	SIPS-PF-15	Sistem dapat menampilkan detail surat keterangan dengan status terverifikasi

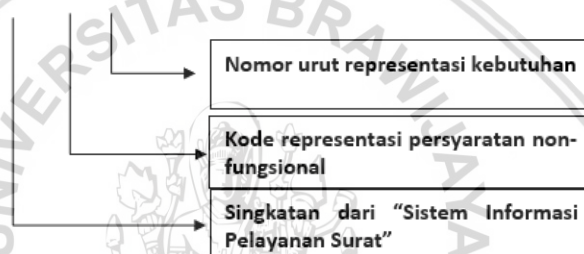
Kode Fitur	Kode Persyaratan Fungsional	Deskripsi
	SIPS-PF-16	Sistem dapat mengganti status surat dari terverifikasi menjadi disetujui
	SIPS-PF-17	Sistem dapat mengganti status surat dari terverifikasi menjadi ditolak
SIPS-FT-06	SIPS-PF-18	Sistem dapat menampilkan daftar surat keterangan dengan status disetujui
	SIPS-PF-19	Sistem dapat menampilkan detail surat keterangan dengan status disetujui
	SIPS-PF-20	Sistem dapat mengganti status surat dari disetujui menjadi selesai
SIPS-FT-07	SIPS-PF-21	Sistem dapat menghapus surat keterangan yang mempunyai status ditolak
SIPS-FT-08	SIPS-PF-22	Sistem dapat menampilkan <i>print preview</i> surat keterangan
	SIPS-PF-23	Sistem dapat mencetak surat keterangan
SIPS-FT-09	SIPS-PF-24	Sistem dapat <i>generate</i> nomor surat secara otomatis saat surat keterangan disimpan
SIPS-FT-10	SIPS-PF-25	Sistem dapat menampilkan isian untuk kata kunci pencarian surat
	SIPS-PF-26	Sistem dapat menampilkan surat keterangan berdasarkan kata kunci yang dimasukkan
SIPS-FT-11	SIPS-PF-27	Sistem dapat menampilkan formulir data pengguna yang telah terisi dengan data pengguna
	SIPS-PF-28	Sistem dapat mengubah data pengguna sesuai dengan data yang telah terganti
SIPS-FT-12	SIPS-PF-29	Sistem dapat menampilkan isian nomor registrasi
	SIPS-PF-30	Sistem dapat menampilkan status surat sesuai dengan nomor registrasi yang dimasukkan
SIPS-FT-13	SIPS-PF-31	Sistem dapat menampilkan isian tanggal awal dan akhir untuk membuat <i>spreadsheet</i> data registrasi surat keterangan

Kode Fitur	Kode Persyaratan Fungsional	Deskripsi
	SIPS-PF-32	Sistem dapat membuat dan mengunduh <i>spreadsheet</i> data registrasi sesuai dengan tanggal awal dan akhir yang telah dimasukkan

4.2.7 Persyaratan Non Fungsional

Persyaratan non-fungsional merupakan kondisi atau kemampuan yang harus dipenuhi oleh sistem supaya fitur sistem dapat berjalan dan dapat memenuhi kebutuhan pengguna. Hubungan antara persyaratan non-fungsional dengan fitur ditunjukkan pada Tabel 4.11. Dibawah ini adalah kodifikasi persyaratan non-fungsional.

Kode : SIPS-NF-xx

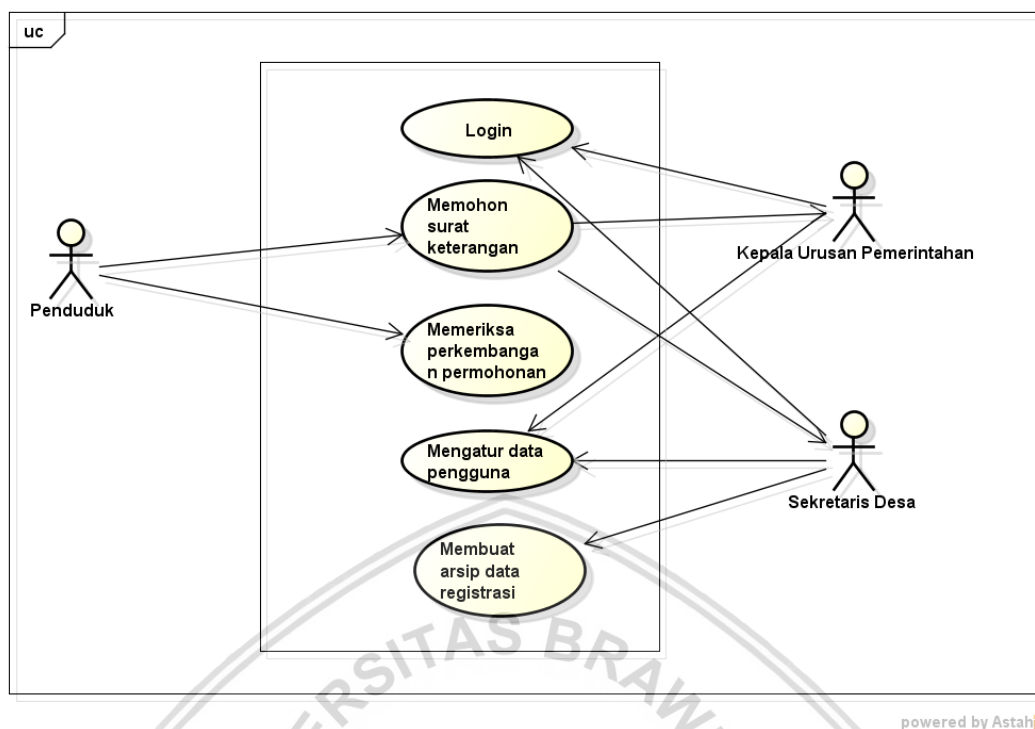


Tabel 4.11 Persyaratan Non Fungsional

Kode Fitur	Kode Persyaratan Non-Fungsional	Deskripsi
SIPS-FT-14	SIPS-NF-01	Sistem dapat diakses pada tempat manapun dan tidak terikat dengan waktu.

4.3 Pemodelan Use Case

Diagram *use case* pada sistem informasi pelayanan surat keterangan yang akan dibuat ditunjukkan pada Gambar 4.3.



Gambar 4.3 Use Case Diagram Sistem Informasi Pelayanan Surat Keterangan Pemerintah Desa Legundi

Setelah tahap analisis persyaratan dilakukan pemodelan use case. Identifikasi aktor, identifikasi *use case*, dan spesifikasi *use case* dilakukan pada pemodelan *use case*. Pada tahap identifikasi aktor dilakukan dengan cara mengelompokkan pengguna sistem berdasarkan karakteristik yang dimiliki. Selanjutnya pada tahap identifikasi *use case* dilakukan dengan menentukan tujuan aktor ketika menggunakan sistem. Tujuan aktor didapatkan dengan menganalisis terhadap hasil identifikasi kebutuhan pengguna pada tahap analisis persyaratan. *Use cases* yang teridentifikasi selanjutnya diberi deskripsi singkat, kondisi yang dibutuhkan sebelum sebuah *use case* dilakukan, urutan tahapan yang harus dilalui oleh *use case* dan kondisi setelah *use case*. Informasi tersebut didokumentasikan ke dalam spesifikasi *use case*.

4.3.1 Use Case Diagram

Hubungan aktor dengan *use case* digambarkan dengan *use case diagram*. *Use case* diidentifikasi dengan cara menentukan tujuan aktor dalam menggunakan sistem. Tujuan aktor dalam menggunakan sistem juga ditentukan berdasarkan fitur-fitur yang diharapkan oleh pengguna. Gambar 4.5 menggambarkan *use case diagram*. Selanjutnya, *use case* yang telah digambarkan dihubungkan dengan solusi yang pada proses bisnis *To-Be* dengan pemangku kepentingan serta dengan fitur yang telah teridentifikasi. *Use case* dihubungkan dengan proses bisnis *To-Be* dilakukan guna menunjukkan hubungan yang lebih baik dalam penggunaan

pemodelan proses bisnis dan *use case*. Tabel 4.12 menunjukkan hubungan *use case* dengan aktivitas proses bisnis.

Tabel 4.12 Hubungan Aktivitas Proses Bisnis *To-Be* dengan *Use Case*

Proses Bisnis	Aktivitas	Use Case
Memohonan Surat Keterangan	Seluruh Aktivitas	Memohon surat keterangan
Periksa Perkembangan Permohonan Surat Keterangan	Seluruh Aktivitas	Memeriksa perkembangan permohonan

Setelah itu *use case* dihubungkan dengan pemangku kepentingan untuk membantu pemangku kepentingan memahami aktivitas yang dapat dilakukan pengguna. Tabel 4.13 terdapat hubungan *use case* dengan pemangku kepentingan.

Tabel 4.13 Hubungan *Use Case* dengan Pemangku Kepentingan

<i>Use Case</i>	Pengguna
Login	Kepala Urusan Pemerintahan, Sekretaris Desa
Memohon surat keterangan	Penduduk, Kepala Urusan Pemerintahan, Sekretaris Desa
Memeriksa perkembangan permohonan	Penduduk
Membuat arsip data registrasi	Sekretaris Desa
Mengatur data pengguna	Kepala Urusan Pemerintahan, Sekretaris Desa

Use case yang teridentifikasi dihubungkan dengan fitur yang teridentifikasi dilakukan untuk mempertegas jika hasil pemodelan *use case* sesuai dengan hasil analisis persyaratan. Tabel 4.14 terdapat hubungan fitur dengan *use case*.

Tabel 4.14 Hubungan *Use Case* dengan Fitur

<i>Use Case</i>	Fitur
Login	SIPS-FT-01
Memohon surat keterangan	SIPS-FT-02, SIPS-FT-03, SIPS-FT-04, SIPS-FT-05, SIPS-FT-06, SIPS-FT-07, SIPS-FT-08, SIPS-FT-09, SIPS-FT-10

Use Case	Fitur
Memeriksa perkembangan permohonan	SIPS-FT-12
Membuat arsip data registrasi	SIPS-FT-13
Mengatur data pengguna	SIPS-FT-11

4.3.2 Deskripsi Aktor

Penjelasan singkat mengenai tanggung jawab aktor yang telah teridentifikasi dijelaskan pada bagian ini. Dalam deskripsi aktor terdapat informasi singkat mengenai tujuan aktor dalam menggunakan sistem informasi. Tabel 4.15 terdapat deskripsi aktor.

Tabel 4.15 Deskripsi Aktor

Nama Aktor	Deskripsi
Penduduk	Aktor Penduduk diperankan oleh penduduk Desa Legundi yang akan menggunakan sistem secara langsung.
Kepala Urusan Pemerintahan	Kepala Urusan Pemerintahan diperankan oleh Kepala Urusan Pemerintahan yang akan mengurus mengenai permohonan surat keterangan
Sekretaris Desa	Sekretaris Desa diperankan oleh Sekretaris Desa yang akan menyetujui surat keterangan.

4.3.3 Spesifikasi Use Case

Pada bagian ini dijelaskan mengenai spesifikasi *use case*. Spesifikasi *use case* adalah penjelasan singkat setiap *use case*. Spesifikasi *use case* juga menjelaskan informasi mengenai aktor yang menjalankan *use case*, kondisi sebelum dan sesudah *use case* dijalankan serta tahapan yang dilakukan *use case*.

4.3.3.1 Login

Penjelasan mengenai spesifikasi dari *use case* Login akan dijelaskan pada Tabel 4.16 yaitu sebagai berikut

Tabel 4.16 Spesifikasi Use Case Login

Login	
Brief Description	Fungsi pada <i>use case</i> ini berguna untuk keamanan informasi pada perangkat lunak yaitu masuk kedalam sistem dengan identitas yang sudah dikenali
Actor	Kepala Urusan Pemerintahan atau Sekretaris Desa

Pre-condition	<ul style="list-style-type: none"> - Perangkat yang digunakan untuk mengakses sistem terhubung dengan internet. - Sistem terhubung dengan server. - Data pengguna tersimpan dalam sistem.
Basic Flow	<p>{use case dimulai}</p> <ol style="list-style-type: none"> 1. <i>Use case</i> dimulai ketika aktor ingin masuk ke dalam sistem. <p>{isi data}</p> <ol style="list-style-type: none"> 2. Aktor mengisi <i>username</i> dan <i>password</i>. 3. Aktor mengirimkan <i>username</i> dan <i>password</i>. 4. Sistem memeriksa isian <i>username</i> dan <i>password</i>. <p>{cek isian}</p> <ol style="list-style-type: none"> 5. Sistem memvalidasi <i>username</i> dan <i>password</i> yang dimasukkan oleh pengguna. <p>{sistem memvalidasi username dan password}</p> <ol style="list-style-type: none"> 6. Sistem menyimpan informasi identitas pengguna. 7. Status login sukses. <p>{use case selesai}</p> <ol style="list-style-type: none"> 8. <i>Use case</i> selesai.
Alternative Flow	<p>A1. Mengatasi Username atau Password kosong</p> <p>Pada {cek isian} jika aktor tidak mengisi salah satu atau kedua isian tersebut maka sistem akan menampilkan pesan <i>error</i> bahwa <i>username</i> dan <i>password</i> wajib untuk diisi, kemudian kembali ke {isi data}.</p> <p>A2. Mengatasi Username atau Password salah</p> <p>Pada {sistem memvalidasi username dan password} apabila <i>username</i> dan <i>password</i> pengguna tidak ditemukan dalam basis data sistem maka sistem akan menampilkan pesan <i>error username dan password salah</i>, kemudian <i>use case</i> selesai.</p>
Post-condition	<ul style="list-style-type: none"> - Identitas aktor teridentifikasi oleh sistem. - Sistem menampilkan informasi sesuai hak akses aktor.

	- Aktor dinyatakan berhasil masuk ke dalam sistem (<i>logged in</i>).
--	---

4.3.3.2 Mengatur Data Pengguna

Penjelasan mengenai spesifikasi dari *use case* mengatur data pengguna akan dijelaskan pada Tabel 4.16 yaitu sebagai berikut:

Tabel 4.17 Spesifikasi *Use Case* Mengatur Data Pengguna

Mengatur Data Pengguna	
Brief Description	Fungsi pada <i>use case</i> ini berguna untuk mengelola pengguna yaitu mengubah data pengguna yang bersangkutan.
Actor	Kepala Urusan Pemerintahan atau Sekretaris Desa
Pre-condition	<ul style="list-style-type: none"> - Perangkat yang digunakan untuk mengakses sistem terhubung dengan internet. - Sistem terhubung dengan server. - Aktor berhasil masuk ke dalam sistem. - Sistem menyimpan data pengguna.
Basic Flow	<p>{use case dimulai}</p> <ol style="list-style-type: none"> 1. <i>Use case</i> dimulai ketika aktor ingin mengubah data diri. 2. Sistem menampilkan formulir pengaturan yang terdiri dari Nama lengkap, <i>Username</i>, <i>Password</i> dan Konfirmasi <i>Password</i> yang telah terisi dengan data diri aktor. <p>{isi data}</p> <ol style="list-style-type: none"> 3. Aktor mengubah data yang ingin diubah. 4. Aktor menyimpan data. 5. Sistem menampilkan konfirmasi perubahan data. <p>{konfirmasi ubah}</p> <ol style="list-style-type: none"> 6. Sistem memeriksa isian data formulir pengaturan. <p>{cek isian}</p> <ol style="list-style-type: none"> 7. Sistem memeriksa <i>username</i> yang dimasukkan pengguna pada basis data untuk memastikan <i>username</i> belum pernah dipakai.

	<p>{sistem memeriksa <i>username</i>}</p> <p>8. Sistem memeriksa kesamaan <i>password</i> dan konfirmasi <i>password</i>.</p> <p>{sistem memeriksa <i>password</i>}</p> <p>9. Sistem mengubah data pengguna pada basis data pengguna.</p> <p>{sistem mengubah data}</p> <p>10. Status ubah data sukses.</p> <p>11. Sistem menampilkan pesan berhasil mengubah data pengguna.</p> <p>{use case selesai}</p> <p>12. Use case selesai.</p>
Alternative Flow	<p>A1. Mengatasi <i>Username</i> dan nama lengkap kosong</p> <p>Pada {cek isian} apabila <i>username</i> atau nama lengkap atau kedua-duanya kosong maka akan menampilkan pesan <i>error</i> disetiap isian, kemudian kembali ke {isi data}.</p> <p>A2. Mengatasi <i>Username</i> telah terpakai</p> <p>Pada {sistem memeriksa <i>username</i>} apabila aktor memasukkan <i>username</i> yang sudah digunakan oleh aktor lain maka sistem akan menampilkan pesan <i>error username</i> sudah terpakai, kemudian kembali ke {pengguna memasukkan data}.</p> <p>A3. Mengatasi <i>password</i> dan konfirmasi <i>password</i> tidak sama</p> <p>Pada {simpan memeriksa <i>password</i>} apabila <i>password</i> dan konfirmasi <i>password</i> yang dimasukkan aktor berbeda maka sistem akan menampilkan pesan <i>error password</i> dan konfirmasi <i>password</i> tidak sama, kemudian kembali ke {pengguna memasukkan data}.</p> <p>A4. Mengatasi Sistem gagal mengubah data pengguna</p> <p>Pada {simpan mengubah data} apabila sistem gagal mengubah data pengguna pada basis data maka sistem akan menampilkan pesan <i>error</i> gagal mengubah data pengguna, kemudian use case selesai.</p> <p>A5. Mengatasi batal mengubah data</p>

	Pada {konfirmasi ubah} apabila aktor batal mengubah data pengguna maka <i>use case</i> selesai.
Post-condition	<ul style="list-style-type: none"> - Aktor dapat melihat formulir pengaturan dan data pengguna yang sudah terganti. - Data pengguna berhasil berubah.

4.3.3.3 Memohon Surat Keterangan

Penjelasan mengenai spesifikasi dari *use case* memohon surat keterangan akan dijelaskan pada Tabel 4.19 yaitu sebagai berikut:

Tabel 4.18 Spesifikasi Use Case Memohon surat keterangan

Memohon Surat Keterangan	
Brief Description	Fungsi pada <i>use case</i> ini berguna untuk permohonan surat keterangan.
Actor	Penduduk, Kepala Urusan Pemerintahan dan Sekretaris Desa
Pre-condition	<ul style="list-style-type: none"> - Perangkat yang digunakan untuk mengakses sistem terhubung dengan internet. - Sistem terhubung dengan server. - Data jenis kelamin, status kawin, jenis surat, bidang surat terisi dalam basis data. - Aktor Kepala Urusan Pemerintahan status login sukses. - Aktor Sekretaris Desa status login sukses.
Basic Flow	<p>{use case dimulai}</p> <ol style="list-style-type: none"> 1. <i>Use case</i> dimulai ketika Penduduk memohon surat keterangan. Jika penduduk tidak datang ke kantor desa maka penduduk membuka halaman utama sistem informasi pelayanan surat keterangan. {permohonan surat} {isi data} 2. Penduduk mengisi salah satu formulir surat keterangan sesuai dengan jenis surat yang dimohon. Formulir surat keterangan terdiri dari surat keterangan belum menikah dari bidang perkawinan, surat keterangan ktp proses dari bidang kependudukan, surat keterangan

	<p>kutipan kelahiran dari bidang kelahiran, surat keterangan kelahiran dari bidang kelahiran dan surat keterangan kemarian dari bidang kematian.</p> <p>3. Penduduk mengirim data surat keterangan. {simpan surat}</p> <p>4. Sistem membuat nomor surat dengan format Kode Bidang/Nomor Urut Surat/404.316.14/Bulan dalam Romawi/Tahun dalam angka. Contoh nomor surat adalah sebagai berikut 474.2/1/404.316.14/IX/2017. Setiap ganti tahun nomor urut surat dimulai dari awal lagi. Kode bidang kelahiran adalah 474.1. Kode bidang perkawinan adalah 474.2. Kode bidang kematian adalah 474.3. Kode bidang kependudukan adalah 470.</p> <p>5. Sistem membuat nomor registrasi dengan format tanggal-kode jenis surat-nomor surat. Contoh REG-2018-04-01-04-1.</p> <p>6. Sistem menyimpan data surat pada basis data sistem dengan status surat diterima. {sistem simpan data}</p> <p>7. Status simpan data surat keterangan berhasil.</p> <p>8. Sistem menampilkan nomor registrasi surat keterangan.</p> <p>9. Jalankan <i>subflow</i> Memverifikasi Surat</p> <p>10. Jalankan <i>subflow</i> Setujui Surat</p> <p>11. Jalankan <i>subflow</i> Mencetak Surat</p> <p>12. Jalankan <i>subflow</i> Menyelesaikan Data</p> <p>13. <i>Use case</i> selesai.</p>
Alternative Flow	<p>A1. Datang ke kantor desa</p> <p>Pada {permohonan surat} apabila penduduk datang ke kantor desa untuk memohon surat keterangan maka menjalankan <i>subflow</i> Datang ke Kantor Desa.</p> <p>A2. Isian kosong dan tidak sesuai format</p> <p>Pada {simpan surat} apabila terdapat isian kosong pada formulir tambah surat maka sistem menampilkan pesan <i>error</i> isian wajib diisi dan apabila isian tidak sesuai dengan format yang ditentukan akan menampilkan pesan <i>error</i>, kemudian kembali ke {isi data}.</p>

	<p>A3. Gagal simpan data surat</p> <p>Pada {sistem simpan data} apabila sistem gagal menyimpan data maka status simpan data surat gagal dan sistem menampilkan pesan simpan data surat gagal, kemudian <i>use case</i> selesai.</p> <p>A4. Mengatasi memilih surat</p> <p>Pada {memilih surat} apabila data surat keterangan terlalu banyak akan menyebabkan kesulitan mencari data maka jalankan <i>subflow</i> Mencari data.</p> <p>A5. Menolak Surat</p> <p>Pada {kepala urusan pemerintahan verifikasi data} atau {sekretaris desa setuju data} atau {kepala urusan pemerintahan selesai data}, apabila kepala urusan pemerintahan/sekretaris desa ingin menolak data maka jalankan <i>subflow</i> Menolak Surat.</p> <p>A6. Menyunting Data</p> <p>Pada {kepala urusan pemerintahan verifikasi data}, apabila kepala urusan pemerintahan ingin menyunting data maka jalankan <i>subflow</i> Menyunting Data.</p> <p>A7. Batal ubah status</p> <p>Pada {konfirmasi ubah status}, apabila ingin membatalkan mengubah status surat maka Kepala urusan pemerintahan/Sekretaris Desa menggunakan fungsi batal dan sistem menampilkan kembali data surat keterangan atau kembali ke {tampil data}.</p> <p>A8. Sistem gagal ubah status</p> <p>Pada {ubah status}, apabila sistem gagal mengubah status surat keterangan maka sistem menampilkan pemberitahuan gagal dan <i>use case</i> selesai.</p> <p>A9. Batal cetak surat</p> <p>Pada {cetak data}, apabila kepala urusan pemerintahan batal mencetak data maka sistem menampilkan surat keterangan dan <i>use case</i> selesai.</p> <p>A10. Menghapus Surat</p>
--	---

	<p>Pada {daftar surat ditolak}, apabila Kepala urusan pemerintahan ingin menghapus data maka jalankan <i>subflow Menghapus Surat</i>.</p> <p>A11. Batal sunting data</p> <p>Pada {konfirmasi sunting data} apabila Kepala urusan pemerintahan batal menyunting data maka kembali ke {tampil sunting data}.</p> <p>A12. Gagal ubah data</p> <p>Pada {sistem ubah data}, apabila sistem gagal mengubah data maka sistem menampilkan pemberitahuan gagal mengubah data kemudian use case selesai.</p> <p>A13. Batal hapus data</p> <p>Pada {konfirmasi hapus} apabila Kepala urusan pemerintahan batal hapus data maka kembali ke {daftar surat ditolak}.</p> <p>A14. Gagal hapus data</p> <p>Pada {sistem hapus surat}, apabila sistem gagal menghapus surat keterangan maka sistem menampilkan pemberitahuan gagal dan <i>use case</i> selesai</p>
Subflow	<p>S1. Memverifikasi Surat</p> <p><i>Subflow</i> ini digunakan untuk memverifikasi data yang dilakukan oleh Kepala urusan pemerintahan</p> <ol style="list-style-type: none"> 1. Ketika kepala urusan pemerintahan ingin memverifikasi data. 2. Sistem menampilkan daftar surat keterangan dengan status diterima dalam bentuk tabel. 3. Kepala urusan pemerintahan memilih salah satu surat keterangan yang ingin diverifikasi. {memilih surat} 4. Sistem menampilkan detail data surat keterangan yang dipilih. {tampil data} 5. Kepala urusan pemerintahan memeriksa data. 6. Kepala urusan pemerintahan memverifikasi data bahwa data tersebut bisa untuk surat keterangan. {kepala urusan pemerintahan verifikasi data}

	<p>7. Sistem menampilkan konfirmasi mengubah status surat. {konfirmasi ubah status}</p> <p>8. Sistem mengubah status surat menjadi terverifikasi. {ubah status}</p> <p>9. Sistem menampilkan pemberitahuan ubah status berhasil.</p> <p>10. Sistem menampilkan daftar surat dengan status terverifikasi dalam bentuk tabel.</p> <p>11. <i>Use case</i> dilanjutkan ke langkah selanjutnya.</p> <p>S2. Setujui Surat</p> <p><i>Subflow</i> ini digunakan untuk menyetujui surat yang telah terverifikasi. Persetujuan dilakukan oleh sekretaris desa.</p> <p>1. Ketika sekretaris desa ingin menyetujui surat keterangan.</p> <p>2. Sistem menampilkan data surat keterangan dengan status terverifikasi dalam bentuk tabel.</p> <p>3. Sekretaris desa memilih salah satu surat keterangan yang ingin disetujui. {memilih surat}</p> <p>4. Sistem menampilkan detail data surat keterangan yang dipilih. {tampil data}</p> <p>5. Sekretaris desa menyetujui data. {sekretaris desa setuju data}</p> <p>6. Sistem menampilkan konfirmasi ubah status surat. {konfirmasi ubah status}</p> <p>7. Sistem mengubah status surat keterangan menjadi disetujui. {ubah status}</p> <p>8. Sistem menampilkan pemberitahuan ubah status berhasil.</p> <p>9. Sistem menampilkan surat keterangan dengan status disetujui dalam bentuk tabel.</p> <p>10. <i>Use case</i> dilanjutkan ke langkah selanjutnya.</p> <p>S3. Mencetak Surat</p> <p><i>Subflow</i> ini digunakan untuk mencetak surat keterangan dengan status selesai.</p>
--	---

	<ol style="list-style-type: none"> 1. Kepala urusan pemerintahan memilih surat yang ingin dicetak dalam daftar surat keterangan dengan status disetujui. {memilih surat} 2. Sistem menampilkan <i>print preview</i> surat keterangan yang dipilih. 3. Kepala urusan pemerintahan mencetak surat keterangan. {cetak data} 4. Sistem mencetak surat keterangan. 5. <i>Use case</i> dilanjutkan ke langkah selanjutnya. <p>S4. Menyelesaikan Surat</p> <p><i>Subflow</i> ini digunakan untuk mengubah status surat selesai yang dilakukan oleh kepala urusan pemerintahan.</p> <ol style="list-style-type: none"> 1. Ketika kepala urusan pemerintahan ingin mengubah status surat menjadi selesai. 2. Sistem menampilkan daftar surat keterangan dengan status disetujui dalam bentuk tabel. 3. Kepala urusan pemerintahan memilih salah satu surat keterangan yang ingin diganti status menjadi selesai. {memilih surat} 4. Sistem menampilkan detail data surat keterangan yang dipilih. {tampil data} 5. Kepala urusan pemerintahan mengubah status menjadi selesai. {kepala urusan pemerintahan selesai data} 6. Sistem menampilkan konfirmasi mengubah status surat. {konfirmasi ubah status} 7. Sistem mengubah status surat menjadi selesai. {ubah status} 8. Sistem menampilkan pemberitahuan ubah status berhasil. 9. Sistem menampilkan daftar surat dengan status selesai dalam bentuk tabel. 10. <i>Use case</i> dilanjutkan ke langkah selanjutnya. <p>S5. Menolak Surat</p>
--	---

	<p><i>Subflow</i> ini digunakan untuk menolak permohonan surat keterangan.</p> <ol style="list-style-type: none"> 1. Ketika kepala urusan pemerintahan/ sekretaris desa ingin menolak surat keterangan. 2. Kepala urusan pemerintahan/sekretaris desa mengisi isian alasan ditolak. 3. Kepala urusan pemerintahan/sekretaris desa menolak surat keterangan. 4. Sistem menampilkan konfirmasi ubah status. {konfirmasi ubah status} 5. Sistem mengubah status data surat keterangan menjadi ditolak. {ubah status} 6. Sistem menampilkan daftar surat keterangan dengan status ditolak dalam bentuk tabel. {daftar surat ditolak} 7. <i>Use case</i> selesai. <p>S6. Menyunting Data</p> <p><i>Subflow</i> ini digunakan untuk mengubah data surat keterangan.</p> <ol style="list-style-type: none"> 1. Ketika Kepala urusan pemerintahan ingin menyunting data surat keterangan. 2. Sistem menampilkan formulir surat keterangan dengan isian data surat keterangan yang dipilih. {tampil sunting data} {isi data} 3. Kepala urusan pemerintahan mengubah data. 4. Kepala urusan pemerintahan menyimpan data. {simpan surat} 5. Sistem menampilkan konfirmasi ubah data surat keterangan. {konfirmasi sunting data} 6. Sistem mengubah data surat keterangan. {sistem ubah data} 7. Sistem menampilkan pemberitahuan ubah data berhasil.
--	--

	<p>8. Kembali ke {tampil data} yang terdapat pada <i>subflow</i> Verifikasi Surat.</p> <p>S7. Hapus Surat</p> <p><i>Subflow</i> ini digunakan untuk menghapus surat keterangan yang terdapat pada daftar surat keterangan dengan status ditolak.</p> <ol style="list-style-type: none"> 1. Ketika kepala urusan pemerintahan ingin menghapus surat. 2. Sistem menampilkan konfirmasi hapus surat. {konfirmasi hapus} 3. Kepala urusan pemerintahan menghapus surat keterangan. 4. Sistem menghapus surat dalam basis data. {sistem hapus surat} 5. Sistem menampilkan pemberitahuan berhasil. 6. Sistem menampilkan data surat keterangan dengan status ditolak dalam tabel yang telah diperbarui. 7. <i>Use case</i> selesai. <p>S8. Mencari data</p> <p><i>Subflow</i> ini digunakan untuk mencari data surat keterangan tertentu.</p> <ol style="list-style-type: none"> 1. Ketika kepala urusan pemerintahan/sekretaris desa ingin mencari data surat keterangan. 2. Kepala urusan pemerintahan/sekretaris desa mengisi isian pencarian sesuai dengan kata kunci tertentu. 3. Sistem menampilkan surat keterangan berdasarkan kata kunci yang dimasukkan. 4. <i>Use case</i> dilanjutkan ke langkah selanjutnya. <p>S9. Datang ke Kantor Desa</p> <p><i>Subflow</i> ini digunakan untuk permohonan surat ketika penduduk datang ke kantor desa.</p> <ol style="list-style-type: none"> 1. Ketika Penduduk datang ke kantor desa. <p>{isi data}</p>
--	--

	<p>2. Kepala urusan pemerintahan mengisi salah satu formulir surat keterangan sesuai dengan jenis surat yang dimohon. Formulir surat keterangan terdiri dari surat keterangan belum menikah dari bidang perkawinan, surat keterangan ktp proses dari bidang kependudukan, surat keterangan kutipan kelahiran dari bidang kelahiran, surat keterangan kelahiran dari bidang kelahiran dan surat keterangan kemarian dari bidang kematian.</p> <p>3. Kepala urusan pemerintahan menyimpan data surat keterangan. {simpan surat}</p> <p>4. Sistem membuat nomor surat dengan format Kode Bidang/Nomor Urut Surat/404.316.14/Bulan dalam Romawi/Tahun dalam angka. Contoh nomor surat adalah sebagai berikut 474.2/1/404.316.14/IX/2017. Setiap ganti tahun nomor urut surat dimulai dari awal lagi. Kode bidang kelahiran adalah 474.1. Kode bidang perkawinan adalah 474.2. Kode bidang kematian adalah 474.3. Kode bidang kependudukan adalah 470.</p> <p>5. Sistem membuat nomor registrasi dengan format tanggal-kode jenis surat-nomor surat. Contoh REG-2018-04-01-04-1.</p> <p>6. Sistem menyimpan data surat pada basis data sistem dengan status surat diterima. {sistem simpan data}</p> <p>7. Status simpan data surat keterangan berhasil.</p> <p>8. Sistem menampilkan nomor registrasi surat keterangan.</p> <p>9. Kembali ke basic flow nomor 9.</p>
Post-condition	<ul style="list-style-type: none"> - Sistem berhasil menyimpan data surat keterangan. - Kepala urusan pemerintahan/Sekretaris Desa bisa melihat daftar surat keterangan. - Penduduk mendapatkan surat keterangan.

4.3.3.4 Memeriksa Perkembangan Permohonan

Penjelasan mengenai spesifikasi dari *use case* memeriksa perkembangan permohonan akan dijelaskan pada Tabel 4.19 yaitu sebagai berikut:

Tabel 4.19 Spesifikasi *Use Case* Memeriksa Perkembangan Permohonan

Memeriksa Perkembangan Permohonan	
Brief Description	Fungsi pada <i>use case</i> ini berguna untuk memeriksa perkembangan permohonan surat keterangan.
Actor	Penduduk
Pre-condition	<ul style="list-style-type: none"> - Perangkat yang digunakan untuk mengakses sistem terhubung dengan internet - Sistem terhubung dengan server - Data surat keterangan tersimpan pada basis data
Basic Flow	<p>{use case dimulai}</p> <ol style="list-style-type: none"> 1. <i>Use case</i> dimulai ketika aktor ingin memeriksa hasil permohonan surat keterangan. 2. Aktor memasukkan nomor registrasi. 3. Aktor memeriksa perkembangan surat keterangan. {aktor cek} 4. Sistem memuat data status surat dengan nomor registrasi yang telah dimasukkan. 5. Sistem menampilkan status registrasi dari surat keterangan. {tampil status} <p>{use case selesai}</p> <ol style="list-style-type: none"> 6. <i>Use case</i> selesai
Alternative Flow	<p>A1. Mengatasi isian kosong</p> <p>Pada {aktor cek} apabila isian kosong maka sistem menampilkan pesan pemberitahuan isian wajib diisi kemudian <i>use case</i> selesai.</p> <p>A2. Mengatasi status ditolak</p> <p>Pada {tampil status} apabila status ditolak maka sistem menampilkan status surat beserta alasan ditolak kemudian <i>use case</i> selesai.</p> <p>A3. Mengatasi nomor registrasi tidak ada</p> <p>Pada {tampil status} apabila nomor registrasi tidak ada maka sistem menampilkan status surat tidak ditemukan kemudian <i>use case</i> selesai.</p>

Post-condition	Aktor dapat melihat status surat keterangan.
-----------------------	--

4.3.3.5 Membuat Arsip Data Registrasi

Penjelasan mengenai spesifikasi dari *use case* membuat arsip data registrasi akan dijelaskan pada Tabel 4.20 yaitu sebagai berikut:

Tabel 4.20 Spesifikasi Use Case Membuat Arsip Data Registrasi

Membuat Arsip Data Registrasi	
Brief Description	Fungsi pada <i>use case</i> ini berguna untuk membuat arsip data registrasi.
Actor	Sekretaris Desa
Pre-condition	<ul style="list-style-type: none"> - Perangkat yang digunakan untuk mengakses sistem terhubung dengan internet - Sistem terhubung dengan server - Aktor berhasil masuk ke dalam sistem sebagai Sekretaris Desa - Data surat keterangan tersimpan pada basis data
Basic Flow	<p>{use case dimulai}</p> <ol style="list-style-type: none"> 1. <i>Use case</i> dimulai ketika aktor ingin membuat <i>spreadsheet</i> data registrasi. 2. Aktor mengisi tanggal awal dan tanggal akhir. 3. Aktor membuat <i>spreadsheet</i> data registrasi. {unduh data} 4. Sistem memuat data registrasi dengan status selesai sesuai dengan tanggal awal dan tanggal akhir yang dimasukkan oleh aktor. 5. Sistem mengkonversi data ke dalam format file <i>spreadsheet</i>. 6. Sistem membuat data registrasi. <p>{use case selesai}</p> <ol style="list-style-type: none"> 7. <i>Use case</i> selesai.
Alternatif Flow	<p>A1. Menangani tanggal awal dan akhir salah</p> <p>Pada {unduh data} apabila tanggal awal lebih besar dari tanggal akhir maka sistem menampilkan pesan</p>

	pemberitahuan tanggal awal tidak boleh lebih besar dari tanggal akhir kemudian <i>use case</i> selesai.
Post-condition	File spreadsheet data registrasi berhasil terunduh.



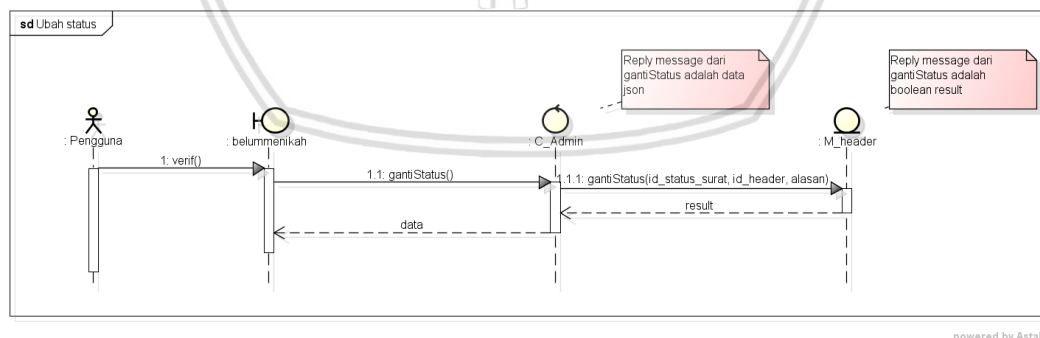
BAB 5 PERANCANGAN

5.1 Perancangan Interaksi Objek

Pada subbab ini dilakukan perancangan interaksi objek dengan cara menggambarkan pertukaran pesan antar entitas aktor, entitas *boundary*, objek *control*, dan objek *model* yang saling berinteraksi. Pemodelan interaksi objek digambarkan ke dalam *sequence diagram*. Sebagian *sequence diagram* yang akan dijelaskan pada bagian ini merupakan penggambaran dari interaksi pada proses simpan surat keterangan, hapus surat keterangan, cek status surat keterangan dan ubah status surat keterangan yang mengacu pada use case Memohon surat keterangan.

5.1.1 Mengubah Status Surat Keterangan

Pada *sequence diagram* mengubah status surat keterangan dimulai ketika pengguna (sekretaris desa dan kepala urusan pemerintahan) sistem ingin mengubah status surat keterangan. Pada sub bab ini diambil proses ubah status terverifikasi yang dilakukan pengguna kepala urusan pemerintahan untuk mengubah status surat menjadi terverifikasi. Dimulai dengan pengguna yaitu kepala urusan pemerintahan menggunakan fungsi *verif*, selanjutnya memanggil fungsi *gantiStatus()* pada *controller* *C_Admin*. Selanjutnya memanggil fungsi *gantiStatus()* pada entitas/*model* *M_header* dengan parameter *id_status_surat* dan *id_header*. Kembalian dari fungsi *gantiStatus* yang terdapat pada *M_header* adalah *boolean result true/false*. Sedangkan *gantiStatus* pada *C_Admin* adalah data json *true/false*. Berikut merupakan *sequence diagram* mengubah status surat keterangan ditunjukkan pada gambar 5.1.

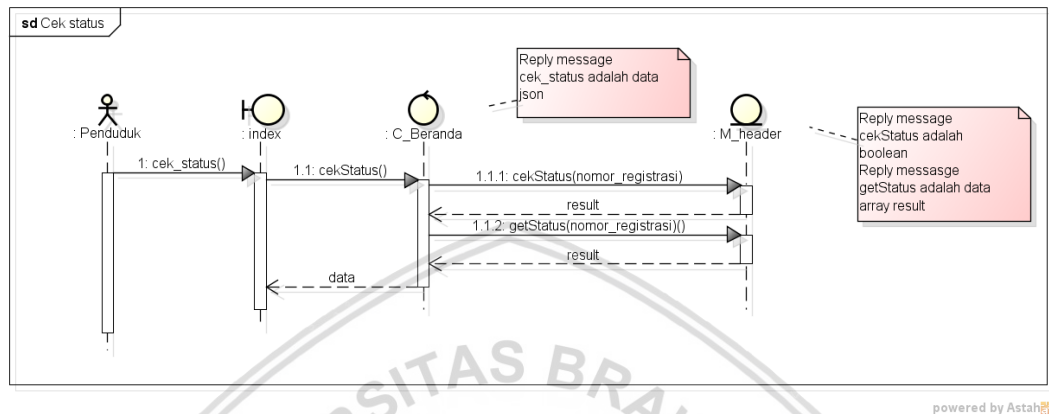


Gambar 5.1 Sequence Diagram Mengubah Status Surat Keterangan

5.1.2 Memeriksa Status Surat Keterangan

Pada *sequence diagram* memeriksa status surat keterangan dimulai ketika pengguna yaitu penduduk sistem menggunakan fungsi cek status pada form cek status surat. Setelah itu data akan dikirimkan menuju *controller* *C_Beranda* dengan memanggil fungsi *cekStatus()*. Tahap selanjutnya adalah memanggil fungsi

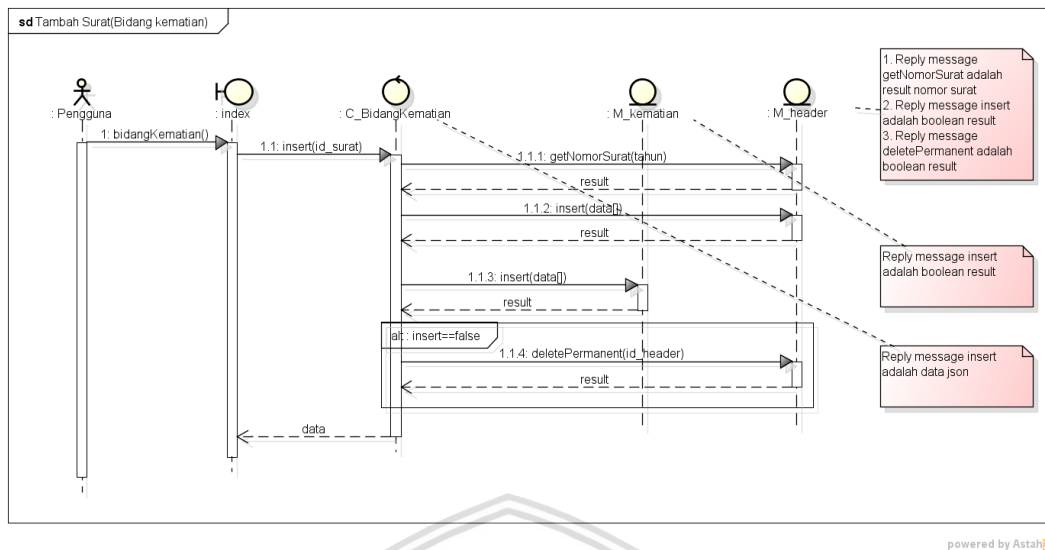
cekStatus() dengan parameter nomor registrasi pada entitas/*model* M_header yang akan menghasilkan data array result berupa status surat dan id surat. Pada cekStatus() *controller* C_Beranda akan menghasilkan data json array berupa status registrasi, dan nomor registrasi serta status berhasil tidaknya proses pemanggilan fungsi cekStatus() pada entitas/*model* M_header. Berikut merupakan *sequence diagram* memeriksa status surat keterangan ditunjukkan pada gambar 5.2.



Gambar 5.2 Sequence Diagram Memeriksa Status Surat Keterangan

5.1.3 Menyimpan Surat Keterangan

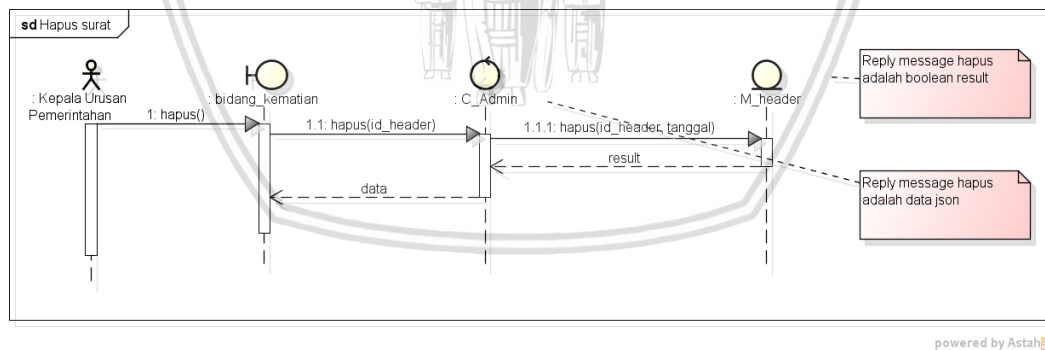
Pada sub bab ini *sequence diagram* menyimpan surat keterangan mengambil dari menyimpan surat keterangan yang terjadi pada bidang kematian dan surat keterangan kematian. Dimulai ketika pengguna (penduduk/kepala urusan pemerintahan) sistem menggunakan fungsi menu Home kemudian mengisi form tambah data surat keterangan pada bidang kematian. Setelah itu data akan dikirimkan menuju *controller* yang C_BidangKematian dengan memanggil fungsi insert dengan parameter id_surat. Setelah itu terjadi pemanggilan fungsi getNomorSurat pada M_header untuk menentukan nomor surat dari surat keterangan yang akan disimpan. Setelah itu *array* data header akan disimpan pada tabel header dengan pemanggilan fungsi insert pada M_header. Selanjutnya *array* data detail akan disimpan pada tabel detail_kematian dengan fungsi *insert* pada M_kematian. Jika *insert* pada M_kematian gagal maka akan menghapus data header yang sebelumnya telah disimpan dengan memanggil deletePermanent pada M_header. Jika *insert* pada M_kematian berhasil akan mengembalikan nilai data. Berikut merupakan *sequence diagram* menambah surat keterangan ditunjukkan pada gambar 5.3.



Gambar 5.3 Sequence Diagram Menyimpan Surat Keterangan

5.1.4 Menghapus Surat Keterangan

Pada *sequence diagram* menghapus surat keterangan diambil dari hapus surat pada bidang kematian. Dimulai dengan pengguna yaitu kepala urusan pemerintahan menggunakan fungsi hapus. Selanjutnya data id_header akan dikirimkan ke *controller* C_BidangKematian selanjutnya *controller* akan memanggil fungsi hapus dengan parameter id_header dan tanggal pada model M_header kemudian akan mengupdate data. Berikut merupakan *sequence diagram* menghapus surat keterangan ditunjukkan pada gambar 5.4.

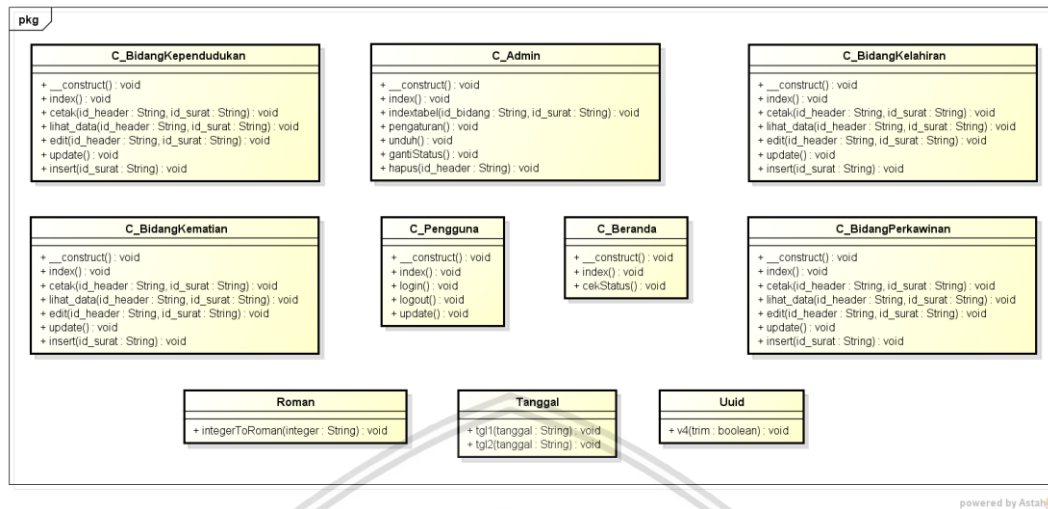


Gambar 5.4 Sequence Diagram Menghapus Surat Keterangan

5.2 Pemodelan Class

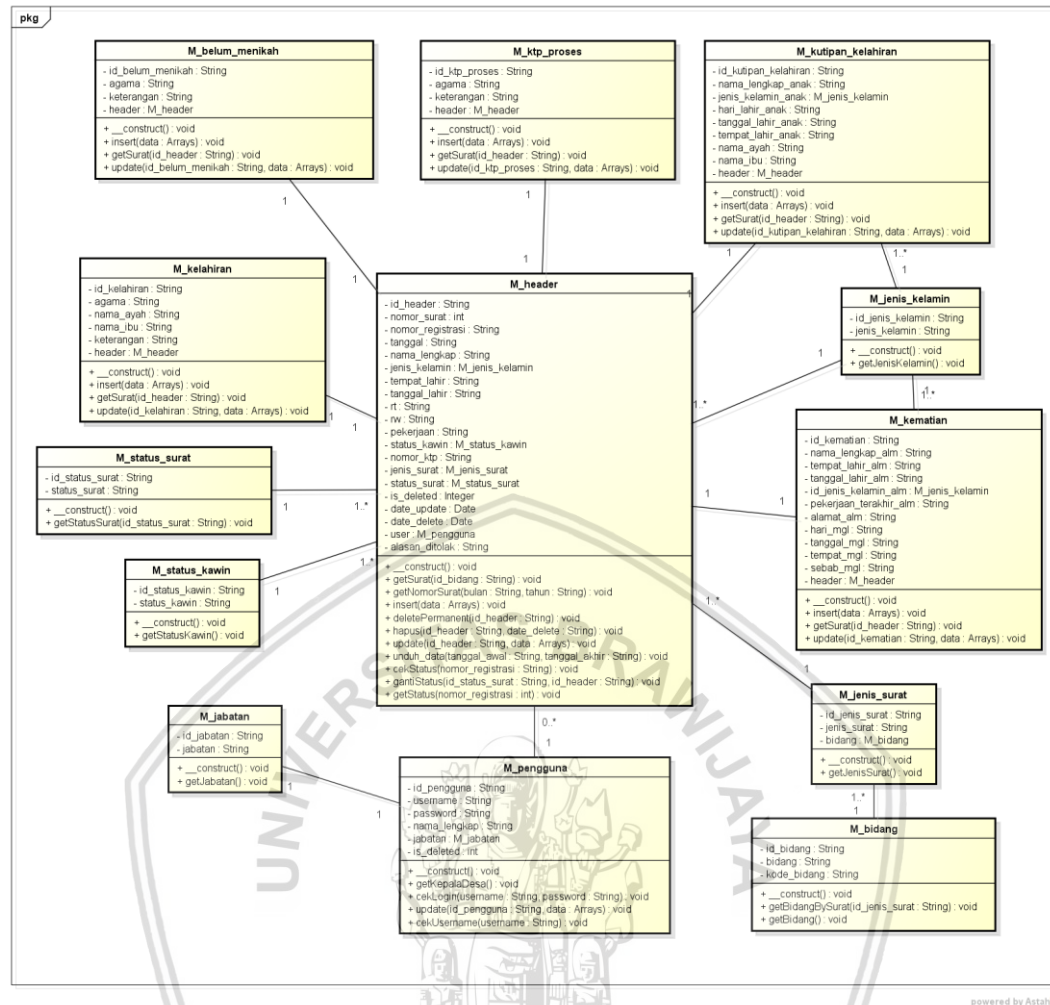
Pada subbab 5.1 telah dijelaskan mengenai hubungan mendetail mengenai perilaku setiap *use case* dengan *sequence diagram*. Tahap selanjutnya adalah mengetahui seluruh hubungan antar *class* yang disertai dengan *atribut* dan *method*/fungsi supaya dalam implementasi perangkat lunak dapat diketahui *class*, atribut serta fungsi yang akan dibuat dan disertai dengan hubungan antar

kelas. *Class diagram* ditunjukkan pada Gambar 5.5, Gambar 5.6 dan Gambar 5.7 sebagai berikut.



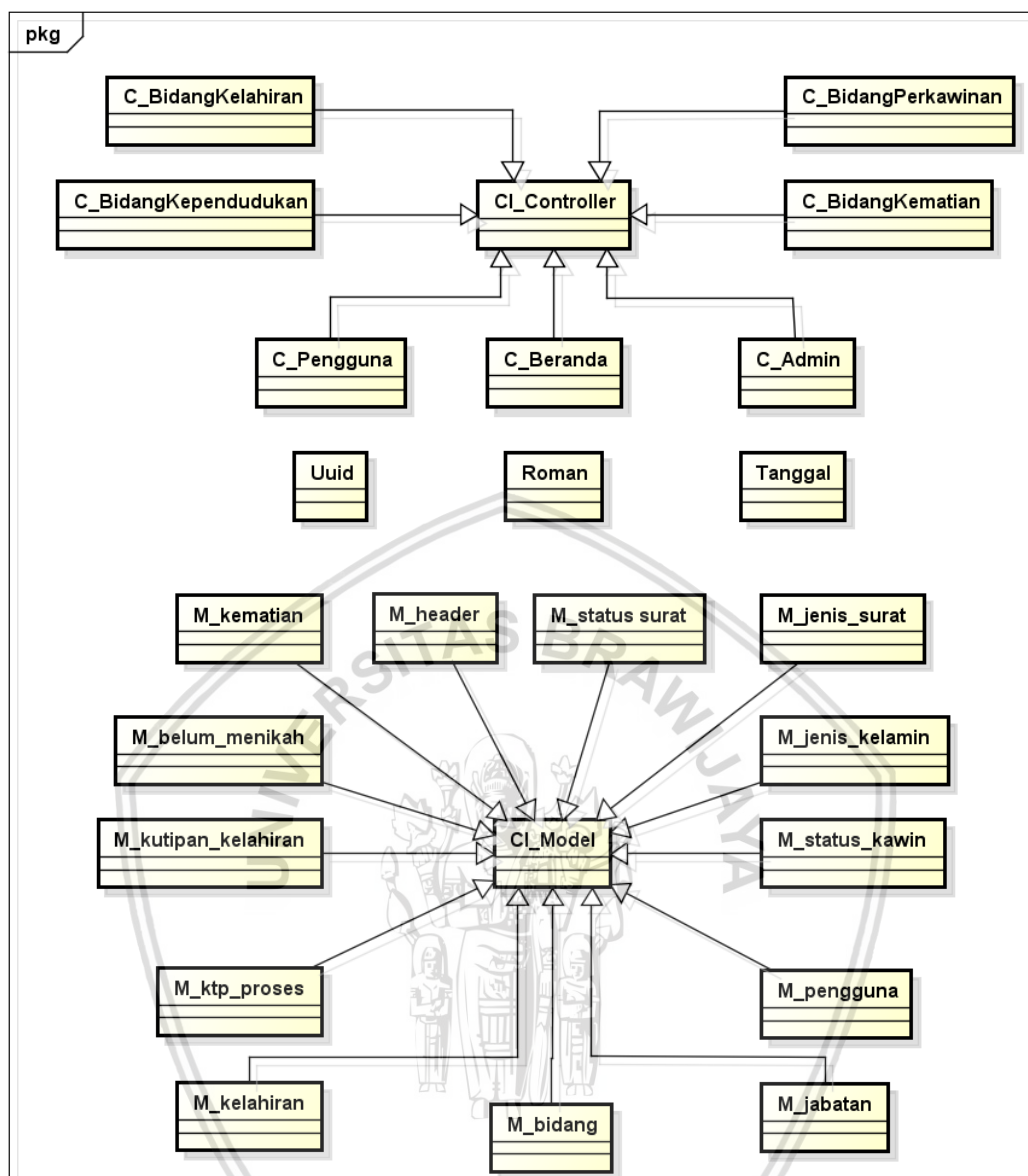
Gambar 5.5 Class diagram untuk implementasi

Pada Gambar 5.5 dijelaskan kelas yang akan digunakan untuk perancangan perangkat lunak. Terdapat 6 kelas *Controller* yaitu C_Admin, C_Beranda, C_User, C_BidangKelahiran, C_BidangPerkawinan, C_BidangKematian dan C_BidangKartuPenduduk. C_Admin digunakan untuk menangani tampilan admin dan semua fungsi yang ada didalamnya seperti tambah data surat keterangan serta untuk menangani kondisi yang bersifat general atau umum. C_Beranda digunakan untuk menangani tampilan beranda untuk penduduk. C_Pengguna digunakan untuk menangani tampilan pengaturan. C_BidangKelahiran digunakan untuk menanganibidang kelahiran beserta fungsi yang ada didalamnya. C_BidangPerkawinan digunakan untuk menangani bidang perkawinan beserta fungsi yang ada didalamnya. C_BidangKematian digunakan untuk menangani bidang kematian beserta fungsi yang ada didalamnya. C_BidangKependudukan digunakan untuk menangani bidang kependudukan beserta fungsi yang ada didalamnya. Roman adalah kelas untuk mengubah angka menjadi huruf roman. Tanggal untuk mengubah format tanggal menjadi berformat Indonesia. UUID adalah kelas untuk membuat uuid versi 4.



Gambar 5.6 Class diagram sebagai domain model

Pada Gambar 5.6 digambarkan kelas entitas yang nantinya akan dijadikan representasi kelas *model* dan juga menjadi acuan untuk pembuatan tabel dalam basis data. Terdapat 12 entitas yaitu M_header yang merepresentasikan data registrasi pada setiap surat beserta data umum pada setiap surat. M_belum_menikah yang merepresentasikan data surat keterangan belum menikah. M_kelahiran yang merepresentasikan data surat keterangan kelahiran. M_kutipan_kelahiran yang merepresentasikan data surat keterangan kutipan kelahiran. M_kematian yang merepresentasikan data surat keterangan kematian. M_ktp_proses yang merepresentasikan data surat keterangan ktp proses. Selanjutnya ada M_pengguna yang merepresentasikan data pengguna. M_jenis_kelamin merepresentasikan data jenis kelamin. M_status_kawin yang merepresentasikan status kawin. M_jabatan yang merepresentasikan jabatan. M_jenis_surat yang merepresentasikan data jenis surat. M_bidang yang merepresentasikan bidang dari surat. M_status_surat yang merepresentasikan status surat keterangan.



powered by Astah

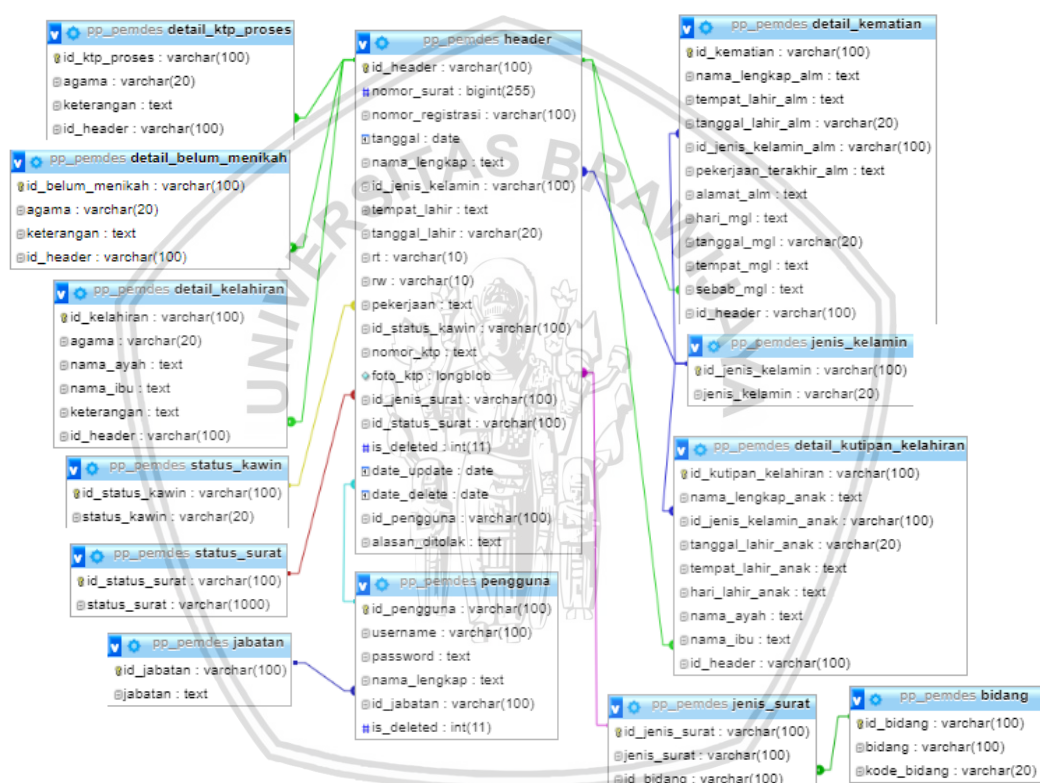
Gambar 5.7 Class diagram dengan kerangka kerja Codeigniter

Gambar 5.7 digambarkan perancangan diagram kelas yang akan diimplementasikan dengan menggunakan kerangka kerja CodeIgniter. Pada perancangan diagram kelas terdapat kelas *CI_Controller* dan *CI_Model* yang berperan sebagai *parent* dari kelas yang dibuat pada tahap perancangan. Kelas *CI_Controller* menjadi *parent* dari class *C_Admin*, *C_Beranda*, *C_BidangKelahiran*, *C_BidangKematian*, *C_BidangKependudukan* dan *C_BidangPerkawinan*. Sedangkan class *CI_Model* menjadi *parent* dari kelas yang pada bab sebelumnya digambarkan sebagai kelas entitas pada implementasinya menjadi kelas *model* yaitu kelas *M_header*, *M_kematian*, *M_kelahiran*, *M_kutipan_kelahiran*, *M_belum_menikah*, *M_ktp_proses*, *M_bidang*, *M_jenis_kelamin*, *M_status_kawin*, *M_jenis_surat*, *M_jabatan*, *M_status_surat* dan *M_pengguna*.

Kelas Uuid adalah kelas *library* yang digunakan untuk *generate* uuid versi 4, kelas Roman adalah kelas *library* untuk mengkonversi angka menjadi romawi dan kelas Tanggal adalah kelas *library* untuk mengkonversi tanggal berformat Indonesia.

5.3 Perancangan Basis Data

Basis data yang digunakan pada sistem informasi pelayanan surat keterangan yang akan dikembangkan menggunakan Mysql. Perancangan basis data dilakukan agar masukan dan keluaran program sesuai dengan yang diharapkan. Perancangan basis data mengacu pada perancangan kelas yang telah dilakukan. Berikut adalah perancangan basis data pada Gambar 5.8.



Gambar 5.8 Physical Data Model Sistem Informasi Pelayanan Surat Keterangan Desa Legundi

Pada PDM Gambar 5.8 dibawah kelas yang menggambarkan domain *model* pada sub bab 5.3 pada Gambar 5.7. Pada gambar diatas terdapat 13 tabel yaitu tabel header yang digunakan untuk menyimpan data registrasi setiap surat keterangan. Tabel detail_kelahiran yang digunakan untuk menyimpan data surat keterangan kelahiran. Tabel detail_kematian yang digunakan untuk menyimpan data surat keterangan kematian. Tabel detail_belum_menikah yang digunakan untuk menyimpan data surat keterangan belum menikah. Tabel detail_kutipan_kelahiran yang digunakan untuk menyimpan data surat keterangan kutipan kelahiran. Tabel detail_ktp_proses yang digunakan untuk

menyimpan data surat keterangan ktp proses. Tabel pengguna yang digunakan untuk menyimpan data pengguna. Tabel jenis_kelamin untuk menyimpan data jenis kelamin. Tabel status_kawin untuk menyimpan data status kawin. Tabel jabatan untuk menyimpan data jabatan pengguna. Tabel jenis_surat untuk menyimpan jenis surat keterangan. Tabel bidang digunakan untuk menyimpan data bidang surat keterangan.

5.3.1 Tabel Header

Tabel Header digunakan untuk menyimpan data utama dari sebuah surat keterangan. Tabel header merepresentasikan data registrasi sebuah surat keterangan. Struktur tabel Header akan dijelaskan pada tabel 5.1.

Tabel 5.1 Struktur Tabel Header

No	Nama	Tipe	Lebar	Keterangan
1	Id_header	Varchar	100	Id header menggunakan uuid dan sebagai <i>primary key</i>
2	Nomor_surat	Varchar	20	Nomor yang digunakan untuk surat
3	Nomor_registrasi	Varchar	100	Nomor registrasi surat keterangan
4	Tanggal	Varchar	100	Tanggal pembuatan
5	Nama_lengkap	Text		Nama lengkap pembuat surat
6	Id_jenis_kelamin	Varchar	100	Id untuk menyimpan jenis kelamin
7	Tempat_lahir	Text		Alamat tempat lahir
8	Tanggal_lahir	Varchar	20	Tanggal lahir pembuat surat
9	Rt	Varchar	10	Nomor RT pembuat surat
10	Rw	Varchar	10	Nomor RW pembuat surat
11	Pekerjaan	Text		Pekerjaan pembuat surat
12	Id_status_kawin	Varchar	100	Id status kawin pembuat surat
13	Nomor_ktp	Text		Nomor KTP pembuat surat
14	Id_jenis_surat	Varchar	100	Jenis surat yang dibuat
15	Id_status_surat	Varchar	100	Status surat keterangan
16	Is_deleted	Int	11	Keterangan data terhapus atau tidak

No	Nama	Tipe	Lebar	Keterangan
17	Date_update	Date		Tanggal <i>update</i> data
18	Date_delete	Date		Tanggal <i>delete</i> data
19	Id_pengguna	Varchar	100	Id pengguna pembuat surat keterangan
20	Alasan_ditolak	Text		Alasan ditolak surat keterangan

5.3.2 Tabel Detail Kutipan Kelahiran

Tabel detail kutipan kelahiran digunakan untuk menyimpan data mengenai surat keterangan kutipan kelahiran. Berikut struktur tabel akan dijelaskan pada tabel 5.2.

Tabel 5.2 Tabel Detail Kutipan Kelahiran

No	Nama	Tipe	Lebar	Keterangan
1	Id_kutipan_kelahiran	Varchar	100	Id menggunakan uuid dan sebagai <i>primary key</i>
2	Nama_lengkap_anak	Text		Nama yang tertera pada surat
3	Id_jenis_kelamin_anak	Varchar	100	Jenis kelamin yang tertera pada surat
4	Hari_lahir_anak	Text		Hari lahir yang tertera pada surat
5	Tanggal_lahir_anak	Varchar	20	Tanggal lahir yang tertera pada surat
6	Tempat_lahir_anak	Text		Alamat tempat lahir
7	Nama_ayah	Text		Nama ayah yang tertera pada surat
8	Nama_ibu	Text		Nama ibu yang tertera pada surat
9	Id_header	Varchar	100	Id untuk mengenali header pembuat surat

5.3.3 Tabel Detail Belum Menikah

Tabel detail belum menikah digunakan untuk menyimpan data surat keterangan belum menikah. Berikut struktur tabel akan dijelaskan pada tabel 5.3.

Tabel 5.3 Tabel Detail Belum Menikah

No	Nama	Tipe	Lebar	Keterangan
1	Id_belum_menikah	Varchar	100	Id menggunakan uuid dan sebagai <i>primary key</i>
2	Agama	Varchar	20	Agama yang tertera pada surat
3	Keterangan	Text		Keterangan yang tertera pada surat
4	Id_header	Varchar	100	Id untuk mengenali header pembuat surat

5.3.4 Tabel Detail Kelahiran

Tabel detail kelahiran digunakan untuk menyimpan data surat keterangan kelahiran. Berikut struktur tabel akan dijelaskan pada tabel 5.4.

Tabel 5.4 Tabel Detail Kelahiran

No	Nama	Tipe	Lebar	Keterangan
1	Id_kelahiran	Varchar	20	Id menggunakan uuid dan sebagai <i>primary key</i>
2	Agama	Varchar	20	Agama yang tertera pada surat
3	Nama_ayah	Text		Nama ayah yang tertera pada surat
4	Nama_ibu	Text		Nama ibu yang tertera pada surat
5	Keterangan	Text		Keterangan yang tertera pada surat
6	Id_header	Varchar	100	Id untuk mengenali header pembuat surat

5.3.5 Tabel Detail Ktp Proses

Tabel detail Ktp proses digunakan untuk menyimpan data surat keterangan Ktp proses. Berikut struktur tabel akan dijelaskan pada tabel 5.5.

Tabel 5.5 Tabel Detail Ktp Proses

No	Nama	Tipe	Lebar	Keterangan
1	Id_ktp_proses	Varchar	100	Id menggunakan uuid dan sebagai <i>primary key</i>

No	Nama	Tipe	Lebar	Keterangan
2	Agama	<i>Varchar</i>	20	Agama yang tertera pada surat
3	Keterangan	<i>Text</i>		Keterangan yang tertera pada surat
4	Id_header	<i>Varchar</i>	100	Id untuk mengenali header pembuat surat

5.3.6 Tabel Detail Kematian

Tabel detail kematian digunakan untuk menyimpan data surat keterangan kematian. Berikut struktur tabel akan dijelaskan pada tabel 5.6.

Tabel 5.6 Tabel Detail Kematian

No	Nama	Tipe	Lebar	Keterangan
1	Id_kematian	<i>Varchar</i>	100	Id menggunakan uuid dan sebagai <i>primary key</i>
2	Nama_lengkap_alm	<i>Text</i>		Nama yang tertera pada surat
3	Tempat_lahir_alm	<i>Text</i>		Alamat tempat lahir
4	Tanggal_lahir_alm	<i>Varchar</i>	20	Tanggal lahir yang tertera pada surat
5	Id_jenis_kelamin_alm	<i>Varchar</i>	100	Jenis kelamin almarhum
6	Pekerjaan_terakhir_alm	<i>Text</i>		Pekerjaan terakhir yang tertera pada surat
7	Alamat_alm	<i>Text</i>		Alamat almarhum
8	Hari_mgl	<i>Text</i>		Hari meninggal yang tertera pada surat
9	Tanggal_mgl	<i>Varchar</i>	20	Tanggal meninggal yang tertera pada surat
10	Tempat_mgl	<i>Text</i>		Tempat meninggal yang tertera pada surat
11	Sebab_mgl	<i>Text</i>		Sebab meninggal almarhum
12	Id_header	<i>Varchar</i>	20	Id untuk mengenali header pembuat surat

5.3.7 Tabel Jabatan

Tabel jabatan digunakan untuk menyimpan data jabatan. Berikut struktur tabel akan dijelaskan pada tabel 5.7.

Tabel 5.7 Tabel Jabatan

No	Nama	Tipe	Lebar	Keterangan
1	Id_jabatan	Varchar	100	Id jabatan sebagai <i>primary key</i>
2	Jabatan	Text		Jabatan pengurus Desa

5.3.8 Tabel Jenis Kelamin

Tabel jenis kelamin digunakan untuk menyimpan data jenis kelamin. Berikut struktur tabel akan dijelaskan pada tabel 5.8.

Tabel 5.8 Tabel Jenis Kelamin

No	Nama	Tipe	Lebar	Keterangan
1	Id_jenis_kelamin	Varchar	100	Id jenis kelamin sebagai <i>primary key</i>
2	Jenis_kelamin	Varchar	20	Jenis kelamin

5.3.9 Tabel Jenis Surat

Tabel jenis surat digunakan untuk menyimpan jenis surat. Berikut struktur tabel akan dijelaskan pada tabel 5.9.

Tabel 5.9 Tabel Jenis Surat

No	Nama	Tipe	Lebar	Keterangan
1	Id_jenis_surat	Varchar	100	Id jenis surat sebagai <i>primary key</i>
2	Jenis_surat	Varchar	20	Jenis surat keterangan
3	Id_bidang	Varchar	100	Id untuk mengenali bidang surat

5.3.10 Tabel Status Kawin

Tabel status kawin digunakan untuk menyimpan status kawin. Berikut struktur tabel akan dijelaskan pada tabel 5.10.

Tabel 5.10 Tabel Status Kawin

No	Nama	Tipe	Lebar	Keterangan
1	Id_status_kawin	Varchar	100	Id status kawin sebagai <i>primary key</i>
2	Status_kawin	Varchar	20	Status Kawin

5.3.11 Tabel Pengguna

Tabel pengguna digunakan untuk menyimpan data pengguna. Berikut struktur tabel akan dijelaskan pada tabel 5.11.

Tabel 5.11 Tabel Pengguna

No	Nama	Tipe	Lebar	Keterangan
1	Id_pengguna	Varchar	100	Id pengguna menggunakan uuid dan sebagai <i>primary key</i>
2	Username	Varchar	100	Penggunaname pengguna
3	Password	Varchar	20	Password pengguna
4	Nama_lengkap	Text		Nama lengkap pengguna
5	Id_jabatan	Varchar	100	Jabatan pengguna
6	Is_deleted	Int	11	Terhapus atau tidak

5.3.12 Tabel Bidang

Tabel bidang digunakan untuk menyimpan data bidang. Berikut struktur tabel akan dijelaskan pada tabel 5.12.

Tabel 5.12 Tabel Bidang

No	Nama	Tipe	Lebar	Keterangan
1	Id_bidang	varchar	100	Id bidang sebagai <i>primary key</i>
2	Bidang	Varchar	100	Bidang surat keterangan
3	Kode_bidang	Varchar	20	Kode bidang surat

5.3.13 Tabel Status Surat

Tabel Status Surat digunakan untuk menyimpan data status surat. Berikut struktur tabel akan dijelaskan pada tabel 5.13.

No	Nama	Tipe	Lebar	Keterangan
1	Id_status_surat	Varchar	100	Id bidang sebagai <i>primary key</i>

No	Nama	Tipe	Lebar	Keterangan
2	Status surat	<i>Varchar</i>	1000	Status surat keterangan

5.4 Perancangan Algoritme

Pada perancangan algoritme dijelaskan algoritme yang dipakai untuk diimplementasikan. Terdapat beberapa algoritme program yang akan dijelaskan dalam sub bab ini yakni algoritme menyimpan data, menghapus data dan menambahkan pengguna.

5.4.1 Menyimpan Surat Keterangan

Pada sub bab ini akan dijelaskan mengenai proses menambahkan surat pada bidang perkawinan. Proses yang pertama kali dilakukan adalah inisialisasi tanggal, setelah itu ada pengecekan nomor surat. Apabila sudah terdapat nomor surat maka nomor surat akan diinisialisasi nomor surat yang sudah ada ditambah 1. Apabila nomor surat belum ada akan diinisialisasi menjadi 1. Selanjutnya adalah pengecekan jenis surat apabila jenis surat = 1 maka akan masuk ke kategori belum menikah. Selanjutnya terdapat pengecekan *captcha*. Setelah itu inisialisasi variabel yang berhubungan dengan tabel header dan detail_belum_menikah. Setelah itu data akan dikirimkan ke class model M_header dan M_belum_menikah. Apabila kembalian dari pemanggilan fungsi *false* maka akan menampilkan *error* pada halaman, sebaliknya akan menampilkan berhasil. Apabila *captcha* salah atau belum diisi juga akan mengembalikan nilai *captcha* berhasil atau tidak. Pada Tabel 5.13 dapat dilihat perancangan algoritme tambah data surat keterangan pada bidang perkawinan.

Tabel 5.13 Algoritme Menyimpan Surat Keterangan

No	Pseudocode
1	Mulai
2	Init datetime ke Asia/Bangkok
3	Init tanggal = date Y-m-d
4	Init query = memanggil fungsi getNomorSurat parameter date year dari M_header
5	FOREACH query as row
6	Init nomor_surat_max ke row->nomor_surat
7	END FOREACH
8	IF nomor-surat_max == null THEN
9	Init nomor_surat = 1
10	ELSE
11	Init nomor_surat = nomor_surat_max +1
12	END IF
13	IF id_surat == 1 THEN
14	IF isset response captcha THEN
15	Init secret key sesuai secret key captcha

```

16 Init verifyresponse = get content dari retun captcha
17 Init responsedata = get json verifyresponse
18 IF responsedata == succes THEN
19   Init id_header = memanggil fungsi v4 dari class uuid
20   Init nomor_registrasi = tanggal."-".id_surat."-".nomor_surat
21   Init nama_lengkap = post(nama_lengkap)
22   Init jenis_kelamin = post(jenis_kelamin)
23   Init tempat_lahir = post(tempat_lahir)
24   Init tanggal_lahir = post(tanggal_lahir)
25   Init rt = post(rt)
25   Init rw = post(rw)
26   Init pekerjaan = post(pekerjaan)
27   Init status_kawin = post(status_kawin)
28   Init nomor_ktp = post(nik)
29   Init id_status_surat = '1'
30   IF isset files(foto_ktp) THEN
31     Init foto_ktp = base64_encode(files_get_content(files(foto_ktp)))
32   ELSE
33     Init foto_ktp == null
34   END IF
35   IF isset session id_pengguna THEN
36     Init id_pengguna == session id_pengguna
37   ELSE
38     Init id_pengguna == null
39   END IF
40   Init array header = array(
    'id_header' => id_header,
    'nomor_registrasi' => nomor_registrasi,
    'nomor_surat' => nomor_surat,
    'tanggal' => tanggal,
    'nama_lengkap' => nama_lengkap,
    'id_jenis_kelamin' => jenis_kelamin,
    'tempat_lahir' => tempat_lahir,
    'tanggal_lahir' => tanggal_lahir,
    'rt' => rt,
    'rw' => rw,
    'pekerjaan' => pekerjaan,
    'id_status_kawin' => status_kawin,
    'nomor_ktp' => nomor_ktp,
    'foto_ktp' => foto_ktp,
    'id_jenis_surat' => id_surat,
    'id_status_surat' => id_status_surat,
    'is_deleted' => 0,
    'date_update' => null,
    'date_delete' => null,
    'id_pengguna' => id_pengguna
  )
41   Init agama = post(agama)
42   Init keterangan = post(keterangan)
43   Init array detail = array(
    'id_belum_menikah' => memanggil fungsi v4 dari class
    uuid,

```

	<pre> 'agama' => agama, 'keterangan' => keterangan, 'id_header' => id_header) 44 Init inserthead = memanggil fungsi insert parameter array header dari M_header 45 IF inserthead == true THEN 46 Init insertdetail = memanggil fungsi insert parameter array detail dari M_belum_menikah 47 IF insertdetail ==true THEN 48 Init data['status'] = true 49 Init data['nomor_registrasi'] = nomor_registrasi 50 Echo json_encode(data) 51 ELSE 52 Init delete = memanggil fungsi deletePermanent parameter id_header dari M_belum_kawin 53 Echo json encode false 54 END IF 55 ELSE 56 Echo json encode false 57 END IF 58 ELSE 59 Echo json encode falsecap 60 END IF 61 ELSE 62 Echo json encode null cap 63 END IF 64 Selesai </pre>
--	---

5.4.2 Menghapus Surat Keterangan

Dalam proses ini akan dijelaskan perancangan algoritme hapus data surat keterangan pada bidang perkawinan yang akan dilakukan oleh sistem. Pada sub bab ini akan dijelaskan proses hapus data surat keterangan pada bidang perkawinan. Saat pengguna menghapus data surat keterangan maka sistem akan memanggil fungsi hapus dengan parameter `id_header` pada class `C_Admin`. Proses yang pertama dilakukan adalah inisialisasi tanggal, setelah itu inisialisasi variabel query dengan pemanggilan fungsi hapus dengan parameter `id_header` dan tanggal pada model `M_header`. Selanjutnya terdapat pengecekan jika *true* maka akan mengirimkan data json *true* dan apabila *false* maka akan mengirimkan data json *false*. Pada Tabel 5.14 dapat dilihat perancangan algoritme hapus data pada bidang perkawinan.

Tabel 5.14 Algoritme Menghapus Surat Keterangan

No	Pseudocode
1	Mulai
2	Init default timezone Asia/Bangkok
3	Init tanggal = date Y-m-d
4	Init query = memanggil fungsi hapus parameter <code>id_header</code> , tanggal dari <code>M_header</code>

5	IF query == true THEN
6	Echo json encode true
7	ELSE
8	Echo json encode false
9	END IF
10	Selesai

5.4.3 Mengubah Status Surat Keterangan

Pada sub bab ini akan dijelaskan perancangan algoritme dari ubah status surat keterangan. Saat surat keterangan berganti status dari diterima menjadi terverifikasi kemudian menjadi disetujui kemudian menjadi selesai atau mungkin ditolak. Yang pertama dilakukan adalah inisialisasi variabel query yang memanggil fungsi gantiStatus dari M_header dengan parameter id_status_surat dan id_header. Kemudian terjadi kondisi jika query *true* maka mengembalikan json *encode true* dan sebaliknya. Tabel 5.15 dapat dilihat perancangan algoritme mengubah status surat keterangan.

Tabel 5.15 Algoritme Mengubah Status Surat Keterangan

No	Pseudocode
1	Mulai
2	Set id_status_surat = post[id_status_surat]
3	Set id_header = post[id_header]
4	Set alasan = post[alasan]
5	Init query = memanggil fungsi gantiStatus parameter id_status_surat, id_header, alasan
6	IF query == true THEN
7	Echo json encode true
8	ELSE
9	Echo json encode false
10	END IF
11	Selesai

5.4.4 Memeriksa Status Surat Keterangan

Dalam proses ini akan dijelaskan perancangan algoritme cek status surat keterangan yang akan dilakukan oleh sistem. Dimulai dengan pengecekan apakah *captcha* diisi tau tidak. Selanjutnya mengambil response dari captcha kemudian memeriksa status nomor registrasi pada basis data dengan memanggil fungsi cekStatus pada M_header dengan parameter nomor registrasi. Setelah itu, mengirimkan data berupa *array* json sebagai *return* data yang berisi status proses, status surat dan nomor registrasi yang telah diambil dari formulir pada antarmuka. Tabel 5.16 dapat dilihat perancangan algoritme memeriksa status surat keterangan.

Tabel 5.16 Algoritme Memeriksa Status Surat Keterangan

No	Pseudocode
1	Mulai

```

2  IF isset post(response) THEN
3      Init secret key captcha
4      Init verifyresponse = get content response captcha
5      Init responsedata = get json verifyresponse
6      IF responsedata==success THEN
7          Init nomor_registrasi = post(nomor_registrasi)
8          Init status = memanggil fungsi cekStatus parameter
          nomor_registrasi dari M_header
9          IF status == true THEN
10             Init surat = memanggil fungsi getStatus parameter nomor
11             registrasi dari M_header
12             FOREACH surat as key
13                 Init status_surat = key->status_surat
14                 Init id_status_surat = key->id_status_surat
15                 Init alasan = key->alasan_ditolak
16             END FOREACH
17             Init data["status"] = true
18             Init data["status_surat"] = $status_surat
19             Init data["nomor_registrasi"] = $nomor_registrasi
20             Init data["alasan"] = $alasan
21             Echo json_encode data
22         ELSE
23             Init data["status"] = true
24             Init data["status_surat"] = 'Tidak ditemukan'
25             Init data["nomor_registrasi"] = $nomor_registrasi
26             Init data["alasan"] = ""
27             Echo json_encode data
28         END IF
29     ELSE
30         Echo json encode falsecap
31     END IF
32 ELSE
33     Echo json encode nullcap
34 END IF
35 Selesai

```

5.5 Perancangan Antarmuka

Pada bagian ini akan menampilkan perancangan antar muka untuk Sistem Informasi Pelayanan Surat Keterangan Desa Legundi.

5.5.1 Login

Pada antarmuka ini digunakan oleh pengguna untuk *login*. Terdapat isian *username* dan *password* serta tombol *login*. Serta terdapat bagian notifikasi untuk memunculkan pemberitahuan. Berikut adalah rancangan antarmuka *login* yang ditunjukkan pada gambar 5.9.

Login

Gambar 5.9 Perancangan Antarmuka *Login*

5.5.2 Admin

Pada antarmuka admin, kepala urusan pemerintahan dapat menyimpan surat dan sekretaris desa dapat membuat arsip data registrasi. Untuk sekretaris desa dapat melakukan persetujuan surat keterangan. Terdapat formulir untuk menyimpan surat serta formulir untuk mengunduh data registrasi serta terdapat menu pada samping kiri halaman. Berikut adalah rancangan antarmuka Admin yang ditunjukkan pada gambar 5.10.

Admin Pelayanan

- Nama
- Bidang Kelahiran
- Bidang Perkawinan
- Bidang Kematian
- Bidang Kartu Penduduk
- Pengaturan
- Logout

Form Tambah Surat Keterangan

Form Unduh Report

Gambar 5.10 Perancangan Antarmuka Admin

5.5.3 Beranda

Pada antarmuka Beranda tertentu terdapat formulir permohonan surat keterangan dan juga formulir cek status registrasi surat keterangan. Antarmuka

beranda ditujukan untuk penduduk. Berikut adalah rancangan antarmuka Beranda yang ditunjukkan pada gambar 5.10.

Gambar 5.11 Perancangan Antarmuka Beranda

5.5.4 Kelola Bidang

Pada antarmuka Kelola Bidang tertentu terdapat tabel yang berisi data registrasi serta aksi yang keberadaannya bergantung pada status surat. Serta terdapat bagian pencarian diatas tabel. Berikut adalah rancangan antarmuka Kelola Bidang yang ditunjukkan pada gambar 5.11.

No	No. Surat	No. Registrasi	Tanggal	No KTP	Nama	Jenis Surat	RT	RW	Action
1	1	Content 1	Content 1	Content 1	Content 1	Content 1	1	1	button

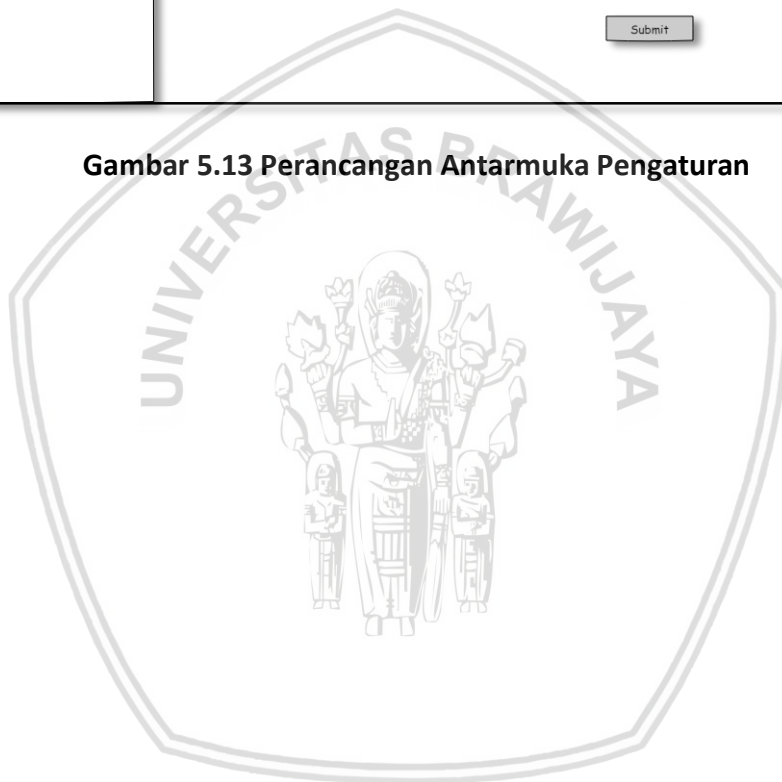
Gambar 5.12 Perancangan Antarmuka Kelola Bidang Kelahiran

5.5.5 Pengaturan

Pada antarmuka Pengaturan terdapat formulir yang digunakan untuk mengubah data pengguna. Berikut adalah rancangan antarmuka Pengaturan yang ditunjukkan pada gambar 5.14.

Admin Pelayanan	
Nama	<div>Pengaturan</div> <div>Form Pengaturan</div> <div>Submit</div>
Bidang Kelahiran	
Bidang Perkawinan	
Bidang Kematian	
Bidang Kartu Penduduk	
Pengaturan	
Logout	

Gambar 5.13 Perancangan Antarmuka Pengaturan



BAB 6 IMPLEMENTASI

Bagian ini akan menjelaskan tentang implementasi yang dilakukan. Terdiri dari sub bab yang menjelaskan spesifikasi lingkungan implementasi, implementasi algoritme dan implementasi antarmuka.

6.1 Spesifikasi Lingkungan Implementasi

Pada bagian ini akan menjelaskan tentang spesifikasi perngkat keras dan perangkat lunak yang digunakan.

6.1.1 Perangkat Keras

Perangkat keras yang digunakan adalah *laptop* HP 14-ac001tx yang memiliki *processor* Intel(R) Core(TM) i5-5200U CPU @ 2.20 GHz kapasitas RAM 4.00 GB, kapasitas *harddisk* 500GB, kartu grafis Intel HD Graphics dan maksimal resolusi layar 1666px x 768px. Tabel 6.1 akan menjelaskan mengenai lingkungan perangkat keras yang digunakan.

Tabel 6.1 Spesifikasi lingkungan perangkat keras

Laptop HP 14-ac001tx	
Processor	Intel(R) Core(TM) i5-5200U CPU @ 2.20 GHz
Kapasitas RAM	4.00 GB
Kapasitas Harddisk	500 GB
Kartu Grafis	Intel HD Graphics
Maksimal Resolusi Layar	1366px x 768px

6.1.2 Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan pada perangkat keras adalah sistem operasi Windows 10 Pro 64-bit, bahasa pemrograman: HTML, PHP, CSS, Javascript, SQL dan perangkat lunak pendukung: XAMPP, Microsoft Visual Studio Code, Google Chrome, Astah, Bizagi Modeler, Microsoft Word 2016, Sublime. Tabel 6.2 menjelaskan lingkungan perangkat lunak yang digunakan.

Tabel 6.2 Spesifikasi lingkungan perangkat lunak

Laptop HP 14-ac001tx	
Sistem Operasi	Windows 10 Pro 64-bit
Bahasa Pemrograman	HTML, PHP, CSS, Javascript, SQL

Perangkat Lunak Pendukung	Xampp Microsoft Visual Studio Code Google Chrome Astar Bizagi Modeler Microsoft Word 2016 Sublime
----------------------------------	---

6.1.3 Minimal Lingkungan *Deployment*

Spesifikasi minimal lingkungan *deployment* sistem dapat dilihat pada tabel dibawah ini. Spesifikasi minimal lingkungan *deployment* untuk sistem adalah kapasitas memori 85 MB, kapasitas RAM 256 MB, versi PHP 5.3 atau lebih dan *library* pendukung php-font-lib v0.3.x.

Tabel 6.3 Spesifikasi Minimal Lingkungan *Deployment*

Kapasitas RAM (Minimal)	256 MB
Kapasitas Memori (Minimal)	85 MB
Versi PHP	5.3 atau lebih
<i>Library</i> pendukung	php-font-lib v0.3.x

6.2 Implementasi Algoritme

Implementasi algoritme adalah kode program yang telah jadi mengacu pada algoritme atau *pseudocode* yang telah dibuat sebelumnya pada tahap perancangan. Berikut ini adalah hasil dari implementasi algoritme yang dikelompokkan ke dalam beberapa bagian yang berkaitan dengan proses Simpan Surat, Hapus Surat, Ubah Status dan Cek Status.

6.2.1 Menyimpan Surat Keterangan

Pada sub bab ini akan dijelaskan mengenai proses menyimpan surat pada bidang perkawinan. Fungsi yang digunakan adalah *insert* dengan parameter *id_surat* yang terdapat pada *controller C_BidangPerkawinan*. Proses yang pertama kali dilakukan adalah inisialisasi tanggal, setelah itu ada pengecekan nomor surat. Apabila sudah terdapat nomor surat maka nomor surat akan diinisialisasi nomor surat yang sudah ada ditambah 1. Apabila nomor surat belum ada akan diinisialisasi menjadi 1. Selanjutnya adalah pengecekan jenis surat apabila jenis surat = 1 maka akan masuk ke kategori belum menikah. Selanjutnya terdapat pengecekan *captcha*. Setelah itu inisialisasi variabel yang berhubungan dengan tabel header dan detail_belum_menikah. Setelah itu data akan dikirimkan ke kelas



model M_header dan M_belum_menikah. Apabila kembalian dari pemanggilan fungsi *false* maka akan menampilkan *error* pada halaman, sebaliknya akan menampilkan berhasil. Apabila *captcha* salah atau belum diisi juga akan mengembalikan nilai *captcha* berhasil atau tidak. Berikut kode program simpan data surat keterangan pada controller bidang perkawinan akan dijabarkan pada Tabel 6.3.

Tabel 6.4 Implementasi Menyimpan Surat Keterangan

Fungsi insert
<pre> \$tanggal = date('Y-m-d'); \$query = \$this->M_header->getNomorSurat(date('Y')); foreach (\$query as \$row){ \$nomor_surat_max = \$row->nomor_surat; } if(\$nomor_surat_max==null){ \$nomor_surat = 1; }else{ \$nomor_surat = \$nomor_surat_max+1; } if(\$id_surat==1){ if(isset(\$_POST['response'])){ \$secret '6LdHMy0UAAAAAAvT3uqEchMALHldwIawuC9TxZJS'; \$arrContextOptions=array("ssl">array("verify_peer">false, "verify_peer_name">false,),); \$verifyResponse file_get_contents('https://www.google.com/recaptcha/api/sitever fy?secret='.\$secret.'&response='.\$_POST['response']); \$responseData = json_decode(\$verifyResponse); if(\$responseData->success){ \$id_header = \$this->uuid->v4(); \$nomor_registrasi = \$tanggal."-".\$id_surat."- ".\$nomor_surat; \$nama_lengkap = \$_POST['nama_lengkap']; \$jenis_kelamin = \$_POST['jenis_kelamin']; \$tempat_lahir = \$_POST['tempat_lahir']; \$tanggal_lahir = \$_POST['tanggal_lahir']; \$rt = \$_POST['rt']; \$rw = \$_POST['rw']; \$pekerjaan = \$_POST['pekerjaan']; \$status_kawin = \$_POST['status_kawin']; \$nomor_ktp = \$_POST['nik']; \$id_status_surat = '1'; if(isset(\$_FILES['foto_ktp']['tmp_name'])&& !empty(\$_FILES['foto_ktp']['tmp_name'])) { </pre>

```

                                $foto_ktp                                =
base64_encode(file_get_contents($_FILES['foto_ktp']['tmp_name']))
);

    }else{
        $foto_ktp = null;
    }

    if(isset($_SESSION['id_pengguna'])){
        $id_pengguna = $_SESSION['id_pengguna'];
    }else{
        $id_pengguna = null;
    }

    $header = array(
        'id_header' => $id_header,
        'nomor_surat' => $nomor_surat,
        'nomor_registrasi' => $nomor_registrasi,
        'tanggal' => $tanggal,
        'nama_lengkap' => $nama_lengkap,
        'id_jenis_kelamin' => $jenis_kelamin,
        'tempat_lahir' => $tempat_lahir,
        'tanggal_lahir' => $tanggal_lahir,
        'rt' => $rt,
        'rw' => $rw,
        'pekerjaan' => $pekerjaan,
        'id_status_kawin' => $status_kawin,
        'nomor_ktp' => $nomor_ktp,
        'foto_ktp' => $foto_ktp,
        'id_jenis_surat' => $id_surat,
        'id_status_surat' => $id_status_surat,
        'is_deleted' => 0,
        'date_update' => null,
        'date_delete' => null,
        'id_pengguna' => $id_pengguna
    );

    $agama = $_POST['agama'];
    $keterangan = $_POST['keterangan'];

    $detail = array(
        'id_belum_menikah' => $this->uuid->v4(),
        'agama' => $agama,
        'keterangan' => $keterangan,
        'id_header' => $id_header
    );

    $insertheader = $this->M_header->
>insert($header);
    if($insertheader==TRUE){
        $insertdetail = $this->M_belum_menikah->
>insert($detail);
        if ($insertdetail==TRUE) {

```



```

                                $data["status"] = true;
                                $data["nomor_registrasi"] =
$nomor_registrasi;

                                echo json_encode($data);
                                }else{
                                    $delete = $this->M_header->deletePermanent($id_header);
                                    echo json_encode(false);
                                }
                                }else{
                                    echo json_encode(false);
                                }
                                }else{
                                    echo json_encode('falsecap');
                                }
                                }else{
                                    echo json_encode('nullcap');
                                }
                                }

```

6.2.2 Menghapus Surat Keterangan

Pada sub bab ini akan dijelaskan implementasi algoritme hapus data surat keterangan pada bidang perkawinan. Saat pengguna menghapus data surat keterangan maka sistem akan memanggil fungsi hapus dengan parameter `id_header` pada kelas `C_Admin`. Proses yang pertama dilakukan adalah inisialisasi tanggal, setelah itu inisialisasi variabel query dengan pemanggilan fungsi hapus dengan parameter `id_header` dan tanggal pada *model* `M_header`. Selanjutnya terdapat pengecekan jika *true* maka akan mengirimkan data json *true* dan apabila *false* maka akan mengirimkan data json *false*. Berikut adalah kode program hapus data pada kelas `C_Admin` yang dapat dilihat pada Tabel 6.4.

Tabel 6.5 Implementasi Menghapus Surat Keterangan

Fungsi hapus
<pre> date_default_timezone_set("Asia/Bangkok"); \$tanggal = date('Y-m-d'); \$query = \$this->M_header->hapus(\$id_header, \$tanggal); if (\$query==true) { echo json_encode(true); }else{ echo json_encode(false); } </pre>

6.2.3 Mengubah Status Surat Keterangan

Pada sub bab ini akan dijelaskan implementasi algoritme dari ubah status surat keterangan. Saat surat keterangan berganti status dari diterima menjadi terverifikasi kemudian menjadi disetujui kemudian menjadi selesai atau mungkin ditolak. Fungsi yang digunakan adalah `gantiStatus` dengan parameter `id_status_surat` dan `id_header` pada `C_Admin`. Yang pertama dilakukan adalah

inisialisasi variabel query yang memanggil fungsi gantiStatus dari M_header dengan parameter id_status_surat, id_header dan alasan. Kemudian terjadi kondisi jika query true maka mengembalikan json *encode true* dan sebaliknya. Berikut adalah kode program ubah status yang akan ditampilkan pada Tabel 6.5.

Tabel 6.6 Implementasi Mengubah Status Surat Keterangan

Fungsi gantiStatus
<pre> \$id_status_surat = \$_POST["status"]; \$id_header = \$_POST["id"]; \$alasan = \$_POST["alasan"]; \$query = \$this->M_header->gantiStatus(\$id_status_surat, \$id_header, \$alasan); if (\$query==true) { echo json_encode(true); }else{ echo json_encode(false); } </pre>

6.2.4 Memeriksa Status Surat Keterangan

Dalam proses ini akan dijelaskan implementasi algoritme cek status surat keterangan yang akan dilakukan oleh sistem. Fungsi yang digunakan adalah cekStatus pada kelas C_Beranda. Dimulai dengan pengecekan apakah diisi atau tidak. Selanjutnya mengambil *response* dari *captcha* kemudian memeriksa status nomor registrasi pada basis data dengan memanggil fungsi cekStatus dan getStatus pada M_header dengan parameter nomor registrasi. Setelah itu, mengirimkan data berupa *array* json sebagai *return* data yang berisi status proses, status surat dan nomor registrasi yang telah diambil dari formulir pada antarmuka.

Tabel 6.7 Implementasi Memeriksa Status Surat Keterangan

Fungsi cekStatus
<pre> if(isset(\$_POST['response'])) { \$secret = '6Lc1d1QUAAAAAakLvEMw6H1wdsqfALA2QPG4qvsv'; \$verifyResponse = file_get_contents('https://www.google.com/recaptcha/api/siteverify?secret='.\$secret.'&response='.\$_POST['response']); \$responseData = json_decode(\$verifyResponse); if(\$responseData->success) { \$nomor_registrasi = \$_POST['nomor_registrasi']; \$status = \$this->M_header->cekStatus(\$nomor_registrasi); if(\$status==true) { \$surat = \$this->M_header->getStatus(\$nomor_registrasi); foreach (\$surat as \$key) { \$status_surat = \$key->status_surat; } } } } </pre>

```

                                $id_status_surat      =      $key->id_status_surat;
>id_status_surat;

                                $alasan = $key->alasan_ditolak;
                                }

                                $data["status"] = true;
                                $data["status_surat"] = $status_surat;

                                $data["nomor_registrasi"] =
$nomor_registrasi;
                                $data["alasan"] = $alasan;
                                echo json_encode($data);
                                }else{
                                $data["status"] = true;
                                $data["status_surat"] = 'Tidak
Ditemukan';
                                $data["nomor_registrasi"] =
$nomor_registrasi;
                                $data["alasan"] = "";
                                echo json_encode($data);
                                }
                                }else{
                                echo json_encode('falsecap');
                                }
                                }else{
                                echo json_encode('nullcap');
                                }

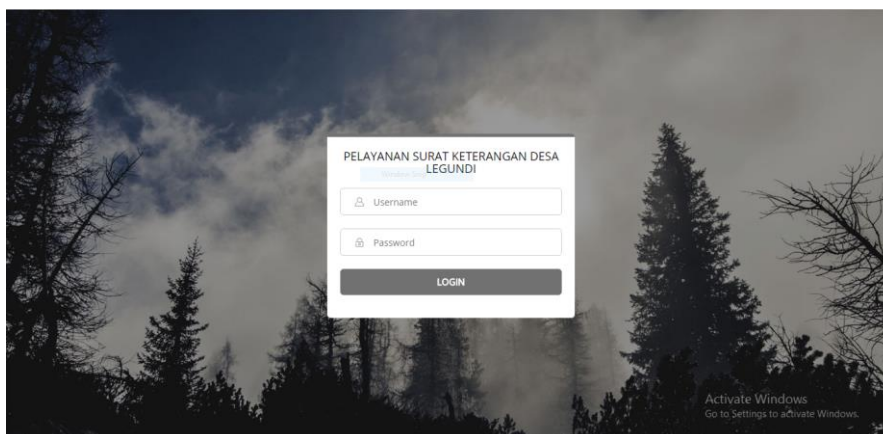
```

6.3 Implementasi Antarmuka

Sub bab implementasi antarmuka berisi beberapa antarmuka yang telah dibuat berdasarkan perancangan antarmuka. Berikut merupakan hasil implementasi antarmuka *Login*, Admin, Beranda, Kelola Bidang Kelahiran dan Pengaturan.

6.3.1 Login

Pada antarmuka ini digunakan oleh pengguna untuk *login*. Terdapat isian *username* dan *password* serta tombol *login*. Serta terdapat bagian notifikasi untuk memunculkan pemberitahuan. Berikut adalah implementasi antarmuka *login* yang ditunjukkan pada gambar 5.9.



Gambar 6.1 Implementasi Antarmuka Login

6.3.2 Admin

Pada antarmuka admin kepala urusan pemerintahan dapat menambah surat dan sekretaris desa dapat membuat arsip data registrasi. Untuk sekretaris desa dapat melakukan persetujuan surat keterangan. Terdapat formulir untuk menambah surat serta formulir untuk mengunduh data registrasi serta terdapat menu pada samping kiri halaman. Berikut adalah implementasi antarmuka Admin yang ditunjukkan pada Gambar 6.3.



Gambar 6.2 Implementasi Antarmuka Admin

6.3.3 Beranda

Pada antarmuka Beranda tertentu terdapat formulir permohonan surat keterangan dan juga formulir cek status registrasi surat keterangan. Antarmuka beranda ditujukan untuk penduduk. Berikut adalah implementasi antarmuka Beranda yang ditunjukkan pada Gambar 6.4.

Pelayanan Surat Keterangan Login

Input Surat Keterangan

Jenis Surat

Pilih Jenis Surat

Cek Status Surat Keterangan

Cek Status Surat Keterangan

Nomor Registrasi

Masukkan Nomor Registrasi

☐ I'm not a robot

Cari

Gambar 6.3 Implementasi Antarmuka Kelola Bidang Kelahiran

6.3.4 Kelola Bidang

Pada antarmuka Kelola Bidang tertentu terdapat tabel yang berisi data registrasi serta aksi yang keberadaannya bergantung pada status surat. Serta terdapat bagian pencarian diatas tabel. Berikut adalah implementasi antarmuka Kelola Bidang yang ditunjukkan pada Gambar 6.4.

BerandaPelayanan

Hi Tio Renny
Online

- Menu Utama
- BidangKelahiran
- BidangPerkawinan
- BidangKependudukan
- BidangKematian
- Pengaturan
- LOGOUT

List Data Bidang Perkawinan
Kode: 474.2
Status: Diterima

Show 10 entries
Search:

No.	No. Surat	No. Registrasi	Tanggal	Nomor KTP	Nama Lengkap	Jenis Surat	RT	RW	Action
1	5	2018-04-25-1-5	04/25/2018	1231233122222222	tio	Keterangan Belum Menikah	12	132	Detail
2	4	2018-04-25-1-10	04/25/2018	1231238132222222	tio	Keterangan Belum Menikah	12	132	Detail

Showing 1 to 2 of 2 entries

Previous
Next

Gambar 6.4 Implementasi Antarmuka Kelola Pengguna

6.3.5 Pengaturan

Pada antarmuka Pengaturan terdapat formulir yang digunakan untuk mengubah data pengguna. Terdapat isian *username*, nama lengkap, *password* baru dan konfirmasi *password* baru. Berikut adalah implementasi antarmuka Pengaturan yang ditunjukkan pada Gambar 6.6.

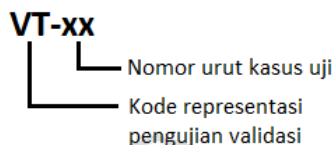
Gambar 6.5 Implementasi Antarmuka Pengaturan



BAB 7 PENGUJIAN

7.1 Pengujian Validasi

Untuk mengetahui sistem dapat berjalan sesuai dengan persyaratan yang telah didefinisikan sebelumnya pengujian validasi dilakukan. Pengujian validasi dilakukan terhadap beberapa fungsi sistem yang digunakan pengguna untuk memohon surat keterangan dan memeriksa perkembangan surat keterangan.



Gambar 7.1 Kodifikasi Pengujian Validasi

Skenario uji yang digunakan dalam pengujian validasi diperoleh dari beberapa alur *use case* Memohon Surat Keterangan dan Memeriksa Perkembangan Surat Keterangan. Setelah itu, kasus uji diidentifikasi berdasarkan skenario uji yang telah ditentukan. Gambar 7.1 menunjukkan kodifikasi kasus uji validasi.

7.1.1 Pengujian Validasi Memohon Surat Keterangan

Pada bagian ini pengujian validasi dilakukan pada beberapa fungsi yang digunakan oleh pengguna untuk memohon surat keterangan melalui sistem. Pada Tabel 7.1 menunjukkan skenario yang berasal dari alur *use-case* Memohon surat keterangan yang digunakan untuk mengidentifikasi kasus uji.

Tabel 7.1 Matriks Skenario Pengujian Validasi Memohon Surat Keterangan

No. Skenario	Alur Awal	Alur Alternatif
Skenario 1	<i>Basic Flow</i>	
Skenario 2	<i>Basic Flow</i>	
Skenario 3	<i>Basic Flow</i>	A10. Menghapus Surat

7.1.1.1 Pengujian Validasi Fungsi Memohon Surat Keterangan

Informasi mengenai rencana pengujian dan kasus uji validasi dengan kode VT-01 ditunjukkan pada Tabel 7.2. Kasus uji dengan kode VT-01 dilakukan untuk memastikan bahwa sistem dapat memohon surat keterangan melalui fungsi menambah surat keterangan dengan kode persyaratan SIPS-PF-06. Diambil sampel menambah atau menyimpan surat keterangan belum menikah. Kasus uji diidentifikasi berdasarkan skenario 1 pada Tabel 7.1. Pengujian validasi fungsi memohon surat keterangan menunjukkan hasil valid.

Tabel 7.2 Rencana Pengujian dan Kasus Uji Validasi Fungsi Memohon Surat Keterangan

Kode Pengujian	VT-01
Nomor Skenario	Skenario 1
Kode Persyaratan	SIKP-PF-06
Tujuan Pengujian	Pengujian untuk memastikan bahwa sistem dapat menambah surat keterangan.
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji berada pada halaman utama sistem. 2. Penguji memilih jenis surat belum menikah. 3. Penguji mengisi formulir surat keterangan belum menikah. 4. Penguji menggunakan fungsi kirim.
Nilai Nama Lengkap	Tio Renndy Winarna
Nilai Jenis Kelamin	Laki-laki
Nilai Tempat Lahir	Ngawi
Nilai Tanggal Lahir	06/07/1996
Nilai RT	08
Nilai RW	01
Nilai Pekerjaan	Pelajar
Nilai Status Kawin	Belum Kawin
Nilai NIK	1234567890123456
Nilai Agama	Islam
Nilai Digunakan Untuk	Melamar pekerjaan
Nilai Foto Ktp	Foto.jpg
Nilai Captcha	True
Hasil yang Diharapkan	Sistem menampilkan nomor registrasi surat keterangan.
Hasil Pengujian	Sistem menampilkan nomor registrasi surat keterangan.
Status Uji	Valid

7.1.1.2 Pengujian Validasi Fungsi Mengubah Status Surat Keterangan

Informasi mengenai rencana pengujian dan kasus uji validasi dengan kode VT-02 ditunjukkan pada Tabel 7.3. Kasus uji dengan kode VT-02 dilakukan untuk memastikan bahwa sistem dapat mengubah status surat keterangan dari diterima menjadi terverifikasi melalui fungsi mengubah status surat keterangan dengan kode persyaratan SIPS-PF-10. Diambil sampel mengubah status surat keterangan belum menikah. Kasus uji diidentifikasi berdasarkan skenario 2 pada Tabel 7.1. Pengujian validasi fungsi mengubah status surat keterangan menunjukkan hasil valid.

Tabel 7.3 Rencana Pengujian dan Kasus Uji Validasi Fungsi Mengubah Status Surat Keterangan

Kode Pengujian	VT-02
Nomor Skenario	Skenario 2
Kode Persyaratan	SIKP-PF-10
Tujuan Pengujian	Pengujian untuk memastikan bahwa sistem dapat mengubah status surat keterangan dari diterima menjadi terverifikasi.
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji teridentifikasi sebagai kepala urusan pemerintahan. 2. Penguji memilih jenis surat belum menikah dengan status surat diterima. 3. Penguji memilih salah satu surat keterangan. 4. Penguji menggunakan fungsi verifikasi.
Hasil yang Diharapkan	Sistem menampilkan status berhasil dan menampilkan daftar surat keterangan dengan status terverifikasi.
Hasil Pengujian	Sistem menampilkan status berhasil dan menampilkan daftar surat keterangan dengan status terverifikasi.
Status Uji	Valid

7.1.1.3 Pengujian Validasi Fungsi Menghapus Surat Keterangan

Informasi mengenai rencana pengujian dan kasus uji validasi dengan kode VT-03 ditunjukkan pada Tabel 7.4. Kasus uji dengan kode VT-03 dilakukan untuk memastikan bahwa sistem dapat menghapus surat keterangan yang mempunyai status ditolak melalui fungsi menghapus surat keterangan dengan kode persyaratan SIPS-PF-21. Diambil sampel menghapus surat keterangan belum menikah. Kasus uji diidentifikasi berdasarkan skenario 3 pada Tabel 7.1. Pengujian validasi fungsi menghapus surat keterangan menunjukkan hasil valid.

Tabel 7.4 Rencana Pengujian dan Kasus Uji Validasi Fungsi Menghapus Surat Keterangan

Kode Pengujian	VT-03
Nomor Skenario	Skenario 3
Kode Persyaratan	SIKP-PF-21
Tujuan Pengujian	Pengujian untuk memastikan bahwa sistem dapat menghapus surat keterangan yang memiliki status ditolak.
Prosedur Uji	<ol style="list-style-type: none"> 1. Penguji teridentifikasi sebagai kepala urusan pemerintahan. 2. Penguji memilih jenis surat belum menikah dengan status surat ditolak. 3. Penguji menggunakan fungsi hapus pada salah satu surat.
Hasil yang Diharapkan	Sistem menampilkan status berhasil dan menampilkan daftar surat keterangan dengan status ditolak.
Hasil Pengujian	Sistem menampilkan status berhasil dan menampilkan daftar surat keterangan dengan status ditolak.
Status Uji	Valid

7.1.2 Pengujian Validasi Memeriksa Perkembangan Permohonan

Pada bagian ini pengujian validasi dilakukan pada beberapa fungsi yang digunakan oleh pengguna untuk memeriksa perkembangan permohonan surat keterangan melalui sistem. Pada Tabel 7.5 menunjukkan skenario yang berasal dari alur *use-case* Periksa perkembangan permohonan yang digunakan untuk mengidentifikasi kasus uji.

Tabel 7.5 Matriks Skenario Pengujian Validasi Memeriksa Perkembangan Surat Keterangan

No. Skenario	Alur Awal	Alur Alternatif
Skenario 1	<i>Basic Flow</i>	
Skenario 2	<i>Basic Flow</i>	A3. Mengatasi nomor registrasi tidak ada

7.1.2.1 Pengujian Validasi Fungsi Memeriksa Status Surat Keterangan

Informasi mengenai rencana pengujian dan kasus uji validasi dengan kode VT-04 ditunjukkan pada Tabel 7.6. Kasus uji dengan kode VT-04 dilakukan guna

memastikan bahwa sistem dapat memeriksa status surat keterangan melalui fungsi memeriksa status surat keterangan dengan kode persyaratan SIPS-PF-30. Skenario 1 yang terdapat pada Tabel 7.5 digunakan untuk kasus uji dibawah ini. Pengujian validasi fungsi memeriksa perkembangan surat keterangan menunjukkan hasil valid.

Tabel 7.6 Rencana Pengujian dan Kasus Uji Validasi Fungsi Memeriksa Perkembangan Surat Keterangan

Kode Pengujian	VT-04
Nomor Skenario	Skenario 1
Kode Persyaratan	SIKP-PF-30
Tujuan Pengujian	Pengujian untuk memastikan bahwa sistem dapat memeriksa perkembangan surat keterangan.
Prosedur Uji	1. Penguji mengisi formulir periksa perkembangan surat keterangan. 2. Penguji menggunakan fungsi cari.
Nilai Registrasi	REG-2018-04-25-4-10 (Nomor registrasi valid/ nomor registrasi yang tersimpan dalam sistem)
Nilai Captcha	True
Hasil yang Diharapkan	Sistem menampilkan nomor registrasi dan status surat keterangan yang sesuai dengan status surat keterangan.
Hasil Pengujian	Sistem menampilkan nomor registrasi dan status surat keterangan yang sesuai dengan status surat keterangan.
Status Uji	Valid

7.1.2.2 Pengujian Validasi Fungsi Memeriksa Status Surat Keterangan

Informasi mengenai rencana pengujian dan kasus uji validasi dengan kode VT-05 ditunjukkan pada Tabel 7.7. Kasus uji VT-05 dilakukan untuk memastikan bahwa sistem dapat memeriksa status surat keterangan melalui fungsi memeriksa status surat keterangan dengan kode persyaratan SIPS-PF-30. Skenario 2 yang terdapat pada Tabel 7.5 digunakan untuk kasus uji dibawah ini. Pengujian validasi fungsi memeriksa perkembangan surat keterangan menunjukkan hasil valid.

Tabel 7.7 Rencana Pengujian dan Kasus Uji Validasi Fungsi Memeriksa Perkembangan Surat Keterangan

Kode Pengujian	VT-05
Nomor Skenario	Skenario 2

Kode Persyaratan	SIKP-PF-30
Tujuan Pengujian	Pengujian untuk memastikan bahwa sistem dapat memeriksa perkembangan surat keterangan.
Prosedur Uji	1. Penguji mengisi formulir periksa perkembangan surat keterangan. 2. Penguji menggunakan fungsi cari.
Nilai Nomor Registrasi	REG-2018-04-25-4-as (Nomor registrasi tidak valid/ nomor registrasi yang tidak tersimpan dalam sistem)
Nilai <i>Captcha</i>	<i>True</i>
Hasil yang Diharapkan	Sistem menampilkan nomor registrasi dan status surat "Tidak Ditemukan".
Hasil Pengujian	Sistem menampilkan nomor registrasi dan status surat "Tidak Ditemukan".
Status Uji	Valid

7.2 Pengujian Responsif

Pengujian responsif dilakukan untuk mengetahui seberapa tanggap sistem apabila dijalankan pada perangkat keras yang mempunyai perbedaan resolusi layar. Pengujian responsif dilakukan karena kebutuhan SIPS-KB-04 dan SIPS-KB-05. Pengujian dilakukan pada 2 perangkat keras. Perangkat keras yang pertama adalah *laptop* HP dengan tipe 14-ac001tx yang mempunyai resolusi layar 1366px × 768px 14 inci. Perangkat keras yang kedua adalah *smartphone* Xiaomi dengan tipe Redmi Note 4 yang mempunyai resolusi layar 1080px × 1920px 5.5 inci. Kedua perangkat tersebut dipilih karena mewakili mayoritas perangkat yang beredar dimasyarakat. Pengujian responsif dilakukan hanya pada halaman beranda dikarenakan halaman beranda adalah halaman yang disediakan untuk mengatasi kebutuhan SIPS-KB-04 dan SIPS-KB-05. Hasil dari pengujian responsif akan dijelaskan dibawah ini.

7.2.1 Resolusi Layar 1366px × 768px 14 inci

Hasil dari pengujian pada perangkat dengan resolusi layar 1366px × 768px 14 inci dapat dilihat pada Gambar 7.2. Dari hasil pengujian yang dilakukan tidak ditemukan elemen pada sistem yang tidak berada pada tempatnya atau rusak dikarenakan resolusi layar. Dapat disimpulkan bahwa pengujian responsif pada perangkat dengan resolusi layar 1366px × 768px 14 inci adalah valid.

Pelayanan Surat Keterangan Login

Input Surat Keterangan

Jenis Surat

Keterangan Belum Menikah

Keterangan Belum Menikah

Nama Lengkap

Masukkan Nama Lengkap

Jenis Kelamin

Pilih Jenis Kelamin

Tempat Lahir

Masukkan Tempat Lahir

Tanggal Lahir

RT

Masukkan RT

RW

Masukkan RW

Pekerjaan

Masukkan Pekerjaan

Status Kawin

Pilih Status Kawin

NIK

Masukkan NIK

Agama

Masukkan Agama

Digunakan Untuk:

Masukkan Keterangan

Unggah Foto KTP (.jpg)

your image

Choose File

No file chosen

or drag a photo

Kirim

Cek Status Surat Keterangan

Nomor Registrasi

Masukkan Nomor Registrasi

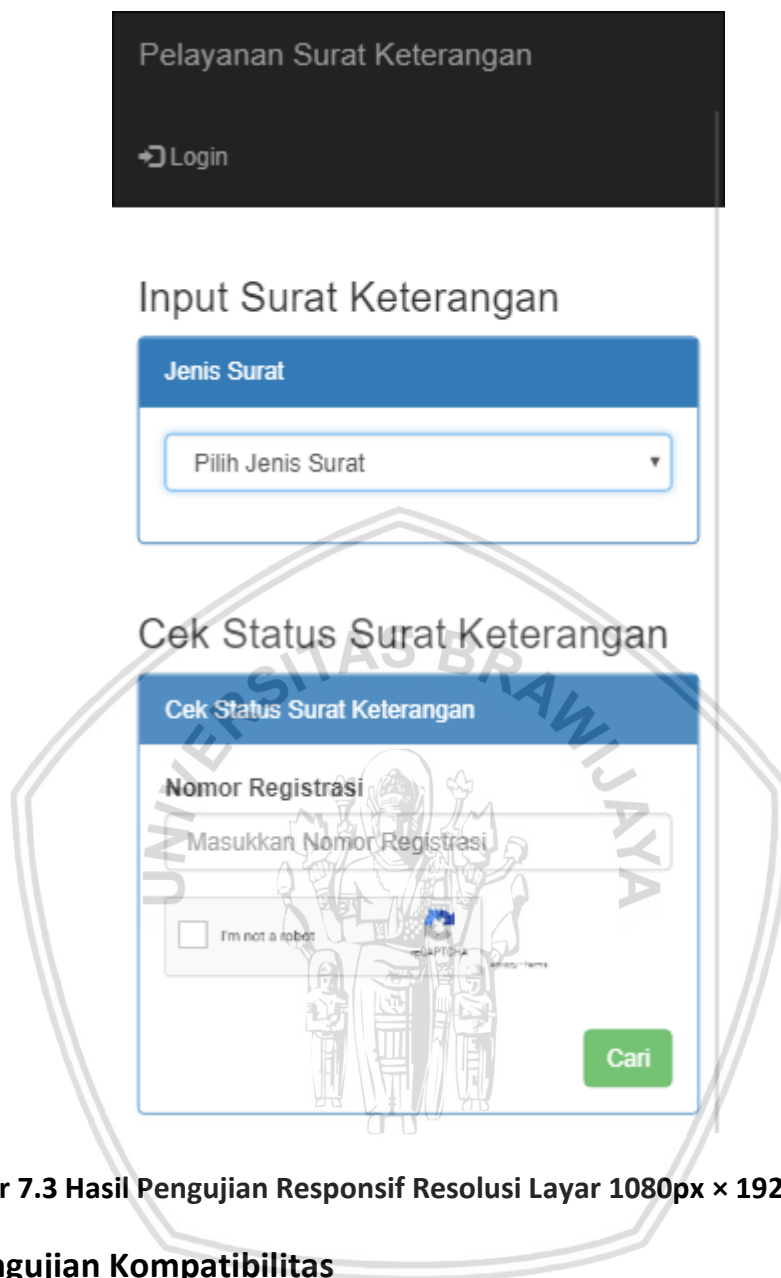
or drag a photo

Cari

Gambar 7.2 Hasil Pengujian Responsif Resolusi Layar 1366px × 768px 14 inci

7.2.2 Resolusi Layar 1080px × 1920px 5.5 inci

Hasil dari pengujian pada perangkat dengan resolusi layar 1080px × 1920px 5.5 inci dapat dilihat pada Gambar 7.3. Dari hasil pengujian yang dilakukan tidak ditemukan elemen pada sistem yang tidak berada pada tempatnya atau rusak dikarenakan resolusi layar. Dapat disimpulkan bahwa pengujian responsif pada perangkat dengan resolusi layar 1080px × 1920px 5.5 inci adalah valid.



Gambar 7.3 Hasil Pengujian Responsif Resolusi Layar 1080px × 1920px 5.5 inci

7.3 Pengujian Kompatibilitas

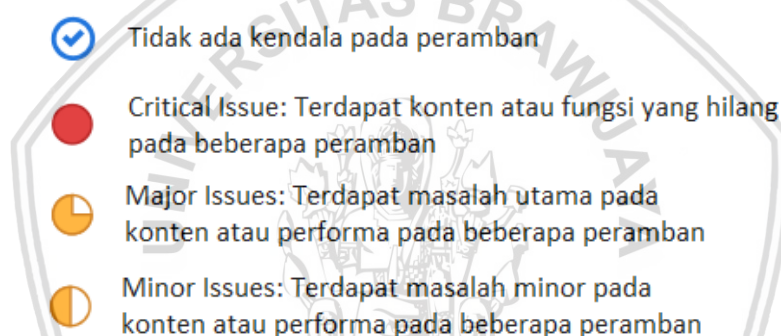
Pengujian kompatibilitas dilakukan untuk mengetahui performa sistem apabila dijalankan pada peramban yang berbeda. Pengujian kompatibilitas dilakukan karena kebutuhan SIPS-KB-04 dan SIPS-KB-05. Perangkat lunak SortSite digunakan untuk menguji kompatibilitas. Perangkat lunak SortSite yang digunakan mempunyai versi 5.29.818.0. Pada Tabel 7.7 terdapat peramban beserta versinya yang digunakan untuk pengujian kompatibilitas.

Tabel 7.8 Peramban yang digunakan pada pengujian kompatibilitas

Nama Peramban	Versi Peramban
Internet Explorer	9,10,11

Nama Peramban	Versi Peramban
Microsoft Edge	16
Firefox	60
Safari	≤ 10, 11
Opera	51
Chrome	66
iOS	≤ 9,10,11
Android	≤ 3,4
Blackberry	≤ 7.1,10.0

Gambar 7.4 menjelaskan kategori masalah pada perangkat lunak SortSite. Ada 3 kategori masalah yaitu *critical issues*, *major issues* dan *minor issues*.



Gambar 7.4 Kategori *compatibility issues* pada SortSite

Hasil pengujian kompatibilitas dapat dilihat pada Gambar 7.5. Hasil pengujian kompatibilitas menunjukkan bahwa Sistem Informasi Pelayanan Surat Keterangan dapat berjalan pada semua peramban yang diuji. Akan tetapi terdapat *Minor Issues* pada peramban Internet Explorer versi 9.

Browser	IE			Edge	Firefox	Safari		Opera	Chrome	iOS			Android		BlackBerry	
Version	9	10	11	16	60	≤ 10	11	51	66	≤ 9	10	11	≤ 3	4*	≤ 7.1	10.0
Critical Issues	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Major Issues	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Minor Issues	⚠	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Gambar 7.5 Hasil Pengujian Kompatibilitas

BAB 8 PENUTUP

8.1 Kesimpulan

Kesimpulan yang dapat diambil berdasarkan penelitian yang telah dilakukan adalah sebagai berikut:

1. Hasil dari pemodelan proses bisnis menghasilkan model proses bisnis usulan yang dapat mempermudah aktivitas permohonan surat keterangan dan aktivitas pemeriksaan perkembangan permohonan surat keterangan pada Pemerintah Desa Legundi. Hal ini karena adanya sistem informasi yang dikembangkan dapat membantu menyelesaikan permasalahan yang ada yaitu masalah yang sudah dijelaskan pada tahap analisis masalah.
2. Analisis persyaratan menghasilkan dokumentasi yang memuat informasi hasil identifikasi pemangku kepentingan, masalah yang dihadapi oleh pemangku kepentingan serta deskripsi kebutuhan pengguna. Setelah itu, kegiatan analisis terhadap kebutuhan pengguna dilakukan sehingga persyaratan sistem dapat dikenali. Persyaratan sistem memuat informasi mengenai 14 fitur dengan fitur utama permohonan surat keterangan, pemeriksaan perkembangan permohonan surat keterangan, pembuatan arsip data registrasi serta pengaturan data pengguna, 32 persyaratan fungsional, dan 1 persyaratan nonfungsional Sistem Informasi Pelayanan Surat Keterangan yang merepresentasikan kemampuan sistem secara umum untuk memohon surat keterangan, memeriksa perkembangan permohonan surat keterangan. Analisis persyaratan juga menghasilkan informasi tentang tujuan penggunaan sistem serta urutan aktivitas yang dilakukan oleh pengguna. Informasi tersebut di dokumentasikan ke dalam 5 diagram *use case*.
3. Hasil dari perancangan sistem yang dilakukan adalah model interaksi objek yang digambarkan ke dalam *sequence diagram*, model objek yang digambarkan ke dalam diagram kelas analisis dan diagram kelas perancangan, perancangan basis data yang digambarkan ke dalam PDM, beberapa rancangan algoritme sistem serta beberapa rancangan antarmuka pengguna. Pada diagram kelas analisis terdapat 13 kelas yang digunakan sebagai acuan yang kemudian menjadi 13 tabel pada perancangan basis data.
4. Hasil pengembangan sistem yang dilakukan berdasarkan hasil perancangan sistem menghasilkan Sistem Informasi Pelayanan Surat Keterangan berbasis web. Sistem Informasi Pelayanan Surat Keterangan dikembangkan dengan menggunakan bahasa pemrograman PHP, HTML, CSS dan Javascript dengan basis data mySql serta kerangka kerja CodeIgniter. Sistem Informasi Pelayanan Surat Keterangan memberikan fitur memohon surat keterangan, memeriksa perkembangan permohonan surat keterangan, membuat arsip data registrasi

dan mengatur data pengguna sebagai kemampuan sistem untuk menyelesaikan kebutuhan pengguna.

5. Hasil pengujian validasi pada sebagian fungsi yang berkaitan dengan memohon surat keterangan dan memeriksa perkembangan permohonan surat keterangan dinyatakan valid. Hasil Pengujian responsif terhadap resolusi layar 1366px × 768px 14 inci dan 1080px × 1920px 5.5 inci dinyatakan valid. Hasil pengujian kompatibilitas menuunjukkan bahwa sistem berjalan baik pada peramban yang menjadi kasus uji. Akan tetapi, terdapat masalah *minor* pada aplikasi peramban Internet Explorer versi 9.

8.2 Saran

Sebagai bahan pertimbangan untuk melakukan pengembangan lebih lanjut pada Sistem Informasi Pelayanan Surat Keterangan saran yang diberikan penulis di antaranya sebagai berikut:

1. Perlu dilakukan penerapan lapangan Sistem Informasi Pelayanan Surat Keterangan pada lokasi studi kasus Pemerintah Desa Legundi. Setelah penerapan lapangan dilakukan perlu adanya kegiatan evaluasi penerapan lapangan sehingga memungkinkan hasil pengembangan lanjut Sistem Informasi Pelayanan Surat Keterangan menjadi lebih baik.
2. Perlu dilakukan pengembangan lanjut Sistem Informasi Pelayanan Surat Keterangan yang dapat mendukung kebutuhan permohonan surat keterangan dengan jenis surat yang lebih banyak atau mendukung pembuatan surat secara dinamis. Perlu dilakukan penambahan fitur *login* yang digunakan oleh penduduk ketika akan memohon surat keterangan untuk keamanan data.
3. Perlu dilakukan evaluasi pada antarmuka pengguna Sistem Informasi Pelayanan Surat Keterangan untuk mengetahui apabila perlu dilakukan perbaikan atau pengembangan lanjut pada antarmuka pengguna.

DAFTAR PUSTAKA

- 1keydata, 2009. *Physical Data Model*. [Online] Available at: <http://www.1keydata.com/datawarehousing/physical-data-model.html> [Accessed 12 April 2018].
- Becker, J., Kugeler, M. & Rosemann, M., 2013. *Process Management: A Guide for the Design of Business Processes*. Berlin: Springer-Verlag.
- Bittner, K. & Spence, I., 2002. *Use Case Modeling*. Boston: Addison Wesley.
- Booch, G. et al., 2007. *Object-Oriented Analysis and Design with Applications*. 3rd ed. Boston: Addison Wesley.
- Booch, G., Rumbaugh, J. & Jacobson, I., 1998. *The Unified Modeling Language User Guide*. Boston: Addison Wesley.
- Cohn, M., 2004. *Advantages of User Stories for Requirements*. [Online] Available at: <https://www.mountaingoatsoftware.com/articles/advantages-of-user-stories-for-requirements> [Accessed 20 May 2018].
- Cohn, M., 2008. *User Stories*. [Online] Available at: <https://www.mountaingoatsoftware.com/agile/user-stories> [Accessed 20 May 2018].
- Group, O. M., 2005. *OMG Unified Modeling Language Specification*. [Online] Available at: <http://www.omg.org/spec/UML/2.0/> [Accessed 03 May 2018].
- Group, O. M., 2005. *Unified Modeling Language: Superstructure*. [Online] Available at: <http://www.omg.org/spec/UML/2.0/Superstructure/PDF/> [Accessed 03 May 2018].
- Group, O. M., 2011. *Business Process Model and Notation (BPMN)*. [Online] Available at: <http://www.omg.org/spec/BPMN/2.0/PDF> [Accessed 03 May 2018].
- Handbook, D. A., 2008. *Agile Business Consortium*. [Online] Available at: <https://www.agilebusiness.org/content/moscow-prioritisation-0> [Accessed 16 April 2018].
- IBM Knowledge Center, 2014. *IBM Knowledge Center*. [Online] Available at: https://www.ibm.com/support/knowledgecenter/en/SS6RBX_11.4.3/com.ibm.sa.bpr.doc/topics/t_Data_obj.html [Accessed 3 September 2017].
- Inc., A., 2013. *Physical Data Model (PDM)s: An Agile Introduction*. [Online] Available at: <http://agilemodeling.com/artifacts/physicalDataModel.htm> [Accessed 12 April 2018].

- Indonesia, R., 2009. *Undang-Undang Republik Indonesia Nomor 25 Tentang Pelayanan Publik*. Jakarta: Sekretariat Negara.
- Laxmi, 2018. *Responsive Web Design Testing: The Complete Beginner's Guide*. [Online] Available at: <https://www.softwaretestinghelp.com/responsive-web-design-testing-guide/> [Accessed 8 June 2018].
- Leffingwell, D. & Widrig, D., 2003. *Managing Software Requirements: A Use Case Approach*. 2nd ed. Boston: Addison Wesley.
- Marakas, G. M. & O'Brian, J., 2013. *Introduction to Information Systems*. 16 ed. New York: McGraw-Hill/Irwin.
- Monk, E. & Wagner, B., 2013. *Concepts in Enterprise Resource Planning*. 4th ed. Boston: Course Technology.
- Pamungkas, A. F., Wardani, N. H. & Saputra, M. C., 2018. Pengembangan Sistem Informasi Persediaan Barang, Harga Pokok Produksi, dan Transaksi Penjualan Berbasis Web (Studi Pada Son Screen Printing Sidoarjo). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, Volume 2, pp. 2075-2084.
- Pressman, R. S., 2010. *Software Engineering A Practitioner's Approach 7th Ed - Roger S. Pressman*. 7th ed. New York: McGraw-Hill.
- Przybyłek, A., 2011. *Bridging the gap between business process models and use-case models*. Gdańsk, Gdańsk University of Technology.
- Sommerville, I., 2011. *Software Engineering Ninth Edition*. New York: Addison-Wesley.
- Sudarevic, B. & Vucsanovic, I. P., 2011. Use of Web Application Frameworks in the Development of Small Applications. *Conference: MIPRO, 2011 Proceedings of the 34th International Convention, Opatija, Croatia, 23-27 May, 2011*.
- Zeaaraoui, A., Bougroun, Z., Belkasmi, M. G. & Bouchentouf, T., 2012. *Object-oriented Analysis and Design Approach for*. Casablanca, Morocco, Second International Conference on the Innovative Computing Technology (INTECH 2012).